

日 本 国 特 許 庁
JAPAN PATENT OFFICE

PCT/JP 2005/000886

25. 1. 2005

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 4 年 1 月 2 9 日
Date of Application:

出 願 番 号 特 願 2 0 0 4 - 0 2 0 8 2 8
Application Number:
[ST. 10/C]: [J P 2 0 0 4 - 0 2 0 8 2 8]

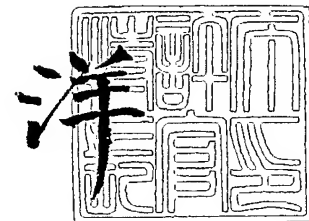
出 願 人 古 庄 晋 二
Applicant(s):



2 0 0 5 年 3 月 3 日

特許庁長官
Commissioner,
Japan Patent Office

小 川



出証番号 出証特 2 0 0 5 - 3 0 1 7 6 4 6

【書類名】 特許願
【整理番号】 PK030064
【あて先】 特許庁長官 殿
【国際特許分類】 G06F 17/30
G06F 15/00
【発明者】
【住所又は居所】 神奈川県横浜市神奈川区松見町 4 丁目 1 1 0 1 番地 7 コートハ
ウス菊名 8 0 4 号
【氏名】 古庄 晋二
【特許出願人】
【識別番号】 598108515
【氏名又は名称】 古庄 晋二
【代理人】
【識別番号】 100099715
【弁理士】
【氏名又は名称】 吉田 聡
【手数料の表示】
【予納台帳番号】 231855
【納付金額】 21,000円
【その他】 国等の委託研究の成果に係る特許出願（平成 1 4 年度 情報処理
振興事業協会 平成 1 4 年度未踏ソフトウェア創造事業「超高速
DB アルゴリズムの超並列環境下のシミュレータ開発」に関する
委託契約）
【提出物件の目録】
【物件名】 特許請求の範囲 1
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1

【書類名】 特許請求の範囲**【請求項 1】**

それぞれ、メモリおよび制御装置を有する、複数のメモリモジュールと、
メモリモジュール間を接続し、あるメモリモジュールの値を他のメモリモジュールに伝達するデータ伝送路とを備え、

各メモリモジュールのメモリが、それぞれ、昇順または降順に重複なく順序付けられた、第 1 の項目の値のリストおよび／または共有化すべき第 2 の項目の値のリストを保持するように構成された情報処理システムであって、

前記各メモリモジュールの制御装置が、

他のメモリモジュールに、前記値のリストに含まれる値を送信するデータ送信手段と、
他のメモリモジュールから、前記値のリストに含まれる値を受信するデータ受信手段と

、
前記データ受信手段により受信された他のメモリモジュールの、前記第 1 の項目の値のリストおよび前記第 2 の項目の値のリストを参照して、他のすべてのメモリモジュールの前記第 1 の項目および第 2 の項目の値のリストに含まれる値を考慮した、共通化された値のリストを生成する共通化手段を備えたことを特徴とする情報処理システム。

【請求項 2】

前記共有化手段が、

前記データ受信手段により受信された他のメモリモジュールの前記第 1 の項目の値のリストを参照して、他のメモリモジュールの、前記第 1 の項目の値のリストに含まれる値を考慮したグローバルな値の順位を決定し、前記グローバルな値の順位を格納するためのグローバル順位格納配列の、自己のメモリモジュールの値に対応する位置に、前記決定された順位を格納する第 1 の順位判定手段と、

前記メモリモジュール自身の第 2 の項目の値のリスト、および、前記データ受信手段により受信された他のメモリモジュールの前記第 2 の項目の値のリストを参照して、前記メモリモジュール自身、および、他のメモリモジュールの、前記第 2 の項目のリストに含まれる値を考慮した、前記第 1 の項目および第 2 の項目が共通化されたグローバルな値の順位を決定し、前記共通化されたグローバルな値の順位を格納するための第 2 のグローバル順位格納配列の、自己のメモリモジュールの値に対応する位置に、前記決定された順位を格納する第 2 の順位判定手段とを有することを特徴とする請求項 1 に記載の情報処理システム。

【請求項 3】

さらに、各メモリモジュールの制御装置が、

前記第 2 の項目の値のリストにおけるそれぞれの値の出現数を格納した第 1 の出現数配列を生成する第 1 の出現数配列生成手段と、

前記全てのメモリモジュールにおける、第 1 の項目の値のリストに関する出現数配列中の出現数に基づき、前記第 1 の出現数配列中の出現数に対応する、前記第 1 の項目の値のリストの値の出現数を格納した第 2 の出現数配列を生成する第 2 の出現数配列生成手段と

、
前記第 2 の出現数配列中の出現数に基づき、前記第 1 の項目の値のリスト中の値を重複させて読み出すデータ読み出し手段とを有することを特徴とする請求項 1 または 2 に記載の情報処理システム。

【請求項 4】

それぞれ、メモリおよび制御装置を有する、複数のメモリモジュールと、メモリモジュール間を接続し、あるメモリモジュールの値を他のメモリモジュールに伝達するデータ伝送路とを備え、各メモリモジュールのメモリが、それぞれ、昇順または降順に重複なく順序付けられた、第 1 の項目の値のリストおよび／または共有化すべき第 2 の項目の値のリストを保持するように構成された情報処理システムにおいて、

前記各メモリモジュールの制御装置において、

他のメモリモジュールに、前記値のリストに含まれる値を送信するデータ送信ステップ

と、

他のメモリモジュールから、前記値のリストに含まれる値を受信するデータ受信ステップと、

前記データ受信手段により受信された他のメモリモジュールの、前記第1の項目の値のリストおよび前記第2の項目の値のリストを参照して、他のすべてのメモリモジュールの前記第1の項目および第2の項目の値のリストに含まれる値を考慮した、共通化された値のリストを生成する共通化ステップとを備えたことを特徴とする情報処理方法。

【請求項5】

前記共有化ステップが、

前記データ受信手段により受信された他のメモリモジュールの前記第1の項目の値のリストを参照して、他のメモリモジュールの、前記第1の項目の値のリストに含まれる値を考慮したグローバルな値の順位を決定し、前記グローバルな値の順位を格納するためのグローバル順位格納配列の、自己のメモリモジュールの値に対応する位置に、前記決定された順位を格納する第1の順位判定ステップと、

前記メモリモジュール自身の第2の項目の値のリスト、および、前記データ受信手段により受信された他のメモリモジュールの前記第2の項目の値のリストを参照して、前記メモリモジュール自身、および、他のメモリモジュールの、前記第2の項目のリストに含まれる値を考慮した、前記第1の項目および第2の項目が共通化されたグローバルな値の順位を決定し、前記共通化されたグローバルな値の順位を格納するための第2のグローバル順位格納配列の、自己のメモリモジュールの値に対応する位置に、前記決定された順位を格納する第2の順位判定ステップとを有することを特徴とする請求項4に記載の情報処理方法。

【請求項6】

さらに、各メモリモジュールの制御装置において、

前記第2の項目の値のリストにおけるそれぞれの値の出現数を格納した第1の出現数配列を生成する第1の出現数配列生成ステップと、

前記全てのメモリモジュールにおける、第1の項目の値のリストに関する出現数配列中の出現数に基づき、前記第1の出現数配列中の出現数に対応する、前記第1の項目の値のリストの値の出現数を格納した第2の出現数配列を生成する第2の出現数配列生成手段と、

前記第2の出現数配列中の出現数に基づき、前記第1の項目の値のリスト中の値を重複させて読み出すデータ読み出しステップとを有することを特徴とする請求項4または5に記載の情報処理方法。

【書類名】明細書

【発明の名称】情報処理システムおよび情報処理方法

【技術分野】

【0001】

本発明は、SIMD (Single Instruction Stream, Multiple Data Stream) を実現可能な並列コンピュータのアーキテクチャを採用した情報処理システムに関する。

【背景技術】

【0002】

社会全体のさまざまな場所にコンピュータが導入され、インターネットをはじめとするネットワークが浸透した今日では、そこそこで、大規模なデータが蓄積されるようになった。このような大規模データを処理するには、膨大な計算が必要で、そのために並列処理を導入しようと試みるのは自然である。

【0003】

並列処理アーキテクチャは「共有メモリ型」と「分散メモリ型」に大別される。前者（「共有メモリ型」）は、複数のプロセッサが1つの巨大なメモリ空間を共有する方式である。この方式では、プロセッサ群と共有メモリ間のトラフィックがボトルネックとなるので、百を越えるプロセッサを用いて現実的なシステムを構築することは容易ではない。したがって、例えば10億個の浮動小数点変数の平方根を計算する際、単一CPUに対する加速比は、せいぜい100倍ということになる。経験的には、30倍程度が上限である。

【0004】

後者（「分散メモリ型」）は、各プロセッサがそれぞれローカルなメモリを持ち、これらを結合してシステムを構築する。この方式では、数百～数万ものプロセッサを組み込んだハードウェアシステムの設計が可能である。したがって、上記10億個の浮動小数点変数の平方根を計算する際の単一CPUに対する加速比を、数百～数万倍とすることが可能である。しかしながら、後者においても、後述するいくつかの課題が存在する。

【特許文献1】国際公開第WO00/10103号パンフレット（第3図および第4図）

【発明の開示】

【発明が解決しようとする課題】

【0005】

【第1の課題：巨大配列の分掌管理】

「分散メモリ型」の第1の課題は、データの分掌管理の問題である。

【0006】

巨大なデータ（一般的には配列なので、以降、配列で説明する）は、1つのプロセッサの所有するローカルメモリに収容できるものではなく、必然的に複数のローカルメモリに分掌管理される。効率的かつ柔軟な分掌管理メカニズムを導入しないと、プログラムの開発および実行に際してさまざまな障害を抱え込むことになることは明らかである。

【0007】

【第2の課題：プロセッサ間通信の効率の低さ】

分散メモリ型システムの各プロセッサが、巨大配列にアクセスしようとする、自己の所有するローカルメモリ上の配列要素に対しては速やかにアクセスできるものの、他のプロセッサが所有する配列要素へのアクセスはプロセッサ間通信を必須とする。このプロセッサ間通信はローカルメモリとの通信に比べ、極端にパフォーマンスが低く、最低でも100クロックかかると言われている。このため、ソート実施時には、巨大配列全域にわたる参照が実施され、プロセッサ間通信が多発するため、パフォーマンスが極端に低下する。

【0008】

この問題点につき、より具体的に説明を加える。1999年現在、パソコンは、1～数個のCPUを用いて、「共有メモリ型」として構成されている。このパソコンに使用される標準的なCPUは、メモリバスの5～6倍程度の内部クロックで動作し、その内部に自

動的な並列実行機能やパイプライン処理機能が装備されており、およそ1データを1クロック（メモリバス）で処理できる。

【0009】

このため、「分散メモリ型」のマルチプロセッサシステムでは、プロセッサ数は多いのに、シングルプロセッサ（共有メモリ型）よりも100倍遅くなることになりかねない。

【0010】

〔第3の課題：プログラムの供給〕

「分散メモリ型」の第3の課題は、多数のプロセッサにどうやってプログラムを供給するか、という問題である。

【0011】

非常に多数のプロセッサに、別々のプログラムをロードし、全体を協調動作させる方式（MIMD: Multiple Instruction Stream, Multiple Data Stream）では、プログラムの作成、コンパイル、配信のために多大な負荷を要する。

【0012】

その一方、多数のプロセッサを同一のプログラムで動作させる方式（SIMD: Single Instruction Stream, Multiple Data Stream）では、プログラムの自由度が減少し、所望の結果をもたらすプログラムが開発できない事態も想定される。

【0013】

本発明は、「分散メモリ型」の上記第1ないし3の課題を解決する方法およびコンピュータアーキテクチャを提供する。

【0014】

ところで、本発明者は、表形式データを記憶するために、項目ごとの情報ブロックを形成し、当該情報ブロックに、項目値を記憶した値リスト、および、当該値リストを指定するための値（ポインタ値）を、レコードごとに記憶したポインタ配列を設け、レコード番号から、ポインタ配列および値リストを順次特定していくことにより、表形式のビューを取得できるような構造および処理方法を考案している（特許文献1参照）。この構造において、レコード数が増大するのにしたがって、上記値リストやポインタ配列、特に、ポインタ配列は非常に大きくなるため、これを、複数のメモリで分掌した上で、単一命令により、検索、集計、ソートなどの処理が実行できるのが望ましい。

【0015】

さらに、データマイニングなど、表形式データを分析する分野をはじめ、多くの分野で、複数の表形式データを、キーとなる項目を共有化することで結合して、結合された新たな表（ビュー）を作成するジョインの技術が必要となっている。したがって、上記特許文献1に記載された構造を用いて、大規模な表形式データを迅速にジョインできるのが望ましい。

【0016】

そこで、本発明は、分散メモリ型において、単一命令により種々のメモリに記憶された配列中の要素を入出力し、処理と通信を統合することで著しく高速な並列処理を実現し、著しく高速な表形式データのジョインが可能な情報処理システムおよび情報処理方法を提供することを目的とする。

【課題を解決するための手段】

【0017】

本発明の目的は、それぞれ、メモリおよび制御装置を有する、複数のメモリモジュールと、メモリモジュール間を接続し、あるメモリモジュールの値を他のメモリモジュールに伝達するデータ伝送路とを備え、各メモリモジュールのメモリが、それぞれ、昇順または降順に重複なく順序付けられた、第1の項目の値のリストおよび／または共有化すべき第2の項目の値のリストを保持するように構成された情報処理システムであって、

前記各メモリモジュールの制御装置が、

他のメモリモジュールに、前記値のリストに含まれる値を送信するデータ送信手段と、他のメモリモジュールから、前記値のリストに含まれる値を受信するデータ受信手段と

、
前記データ受信手段により受信された他のメモリモジュールの、前記第1の項目の値のリストおよび前記第2の項目の値のリストを参照して、他のすべてのメモリモジュールの前記第1の項目および第2の項目の値のリストに含まれる値を考慮した、共通化された値のリストを生成する共通化手段を備えたことを特徴とする情報処理システムにより達成される。

【0018】

好ましい実施態様においては、前記共有化手段が、

前記データ受信手段により受信された他のメモリモジュールの前記第1の項目の値のリストを参照して、他のメモリモジュールの、前記第1の項目の値のリストに含まれる値を考慮したグローバルな値の順位を決定し、前記グローバルな値の順位を格納するためのグローバル順位格納配列の、自己のメモリモジュールの値に対応する位置に、前記決定された順位を格納する第1の順位判定手段と、

前記メモリモジュール自身の第2の項目の値のリスト、および、前記データ受信手段により受信された他のメモリモジュールの前記第2の項目の値のリストを参照して、前記メモリモジュール自身、および、他のメモリモジュールの、前記第2の項目のリストに含まれる値を考慮した、前記第1の項目および第2の項目が共通化されたグローバルな値の順位を決定し、前記共通化されたグローバルな値の順位を格納するための第2のグローバル順位格納配列の、自己のメモリモジュールの値に対応する位置に、前記決定された順位を格納する第2の順位判定手段とを有する。

【0019】

別の好ましい実施態様においては、さらに、各メモリモジュールの制御装置が、

前記第2の項目の値のリストにおけるそれぞれの値の出現数を格納した第1の出現数配列を生成する第1の出現数配列生成手段と、

前記全てのメモリモジュールにおける、第1の項目の値のリストに関する出現数配列中の出現数に基づき、前記第1の出現数配列中の出現数に対応する、前記第1の項目の値のリストの値の出現数を格納した第2の出現数配列を生成する第2の出現数配列生成手段と

、
前記第2の出現数配列中の出現数に基づき、前記第1の項目の値のリスト中の値を重複させて読み出すデータ読み出し手段とを有する。

【0020】

また、本発明の目的は、それぞれ、メモリおよび制御装置を有する、複数のメモリモジュールと、メモリモジュール間を接続し、あるメモリモジュールの値を他のメモリモジュールに伝達するデータ伝送路とを備え、各メモリモジュールのメモリが、それぞれ、昇順または降順に重複なく順序付けられた、第1の項目の値のリストおよび／または共有化すべき第2の項目の値のリストを保持するように構成された情報処理システムにおいて、

前記各メモリモジュールの制御装置において、

他のメモリモジュールに、前記値のリストに含まれる値を送信するデータ送信ステップと、

他のメモリモジュールから、前記値のリストに含まれる値を受信するデータ受信ステップと、

前記データ受信手段により受信された他のメモリモジュールの、前記第1の項目の値のリストおよび前記第2の項目の値のリストを参照して、他のすべてのメモリモジュールの前記第1の項目および第2の項目の値のリストに含まれる値を考慮した、共通化された値のリストを生成する共通化ステップとを備えたことを特徴とする情報処理方法により達成される。

【0021】

好ましい実施態様においては、前記共有化ステップが、

前記データ受信手段により受信された他のメモリモジュールの前記第1の項目の値のリストを参照して、他のメモリモジュールの、前記第1の項目の値のリストに含まれる値を

考慮したグローバルな値の順位を決定し、前記グローバルな値の順位を格納するためのグローバル順位格納配列の、自己のメモリモジュールの値に対応する位置に、前記決定された順位を格納する第1の順位判定ステップと、

前記メモリモジュール自身の第2の項目の値のリスト、および、前記データ受信手段により受信された他のメモリモジュールの前記第2の項目の値のリストを参照して、前記メモリモジュール自身、および、他のメモリモジュールの、前記第2の項目のリストに含まれる値を考慮した、前記第1の項目および第2の項目が共通化されたグローバルな値の順位を決定し、前記共通化されたグローバルな値の順位を格納するための第2のグローバル順位格納配列の、自己のメモリモジュールの値に対応する位置に、前記決定された順位を格納する第2の順位判定ステップとを有する。

【0022】

別の好ましい実施態様においては、さらに、各メモリモジュールの制御装置において、前記第2の項目の値のリストにおけるそれぞれの値の出現数を格納した第1の出現数配列を生成する第1の出現数配列生成ステップと、

前記全てのメモリモジュールにおける、第1の項目の値のリストに関する出現数配列中の出現数に基づき、前記第1の出現数配列中の出現数に対応する、前記第1の項目の値のリストの値の出現数を格納した第2の出現数配列を生成する第2の出現数配列生成手段と

、
前記第2の出現数配列中の出現数に基づき、前記第1の項目の値のリスト中の値を重複させて読み出すデータ読み出しステップとを有する。

【発明の効果】

【0023】

本発明によれば、分散メモリ型において、単一命令により種々のメモリに記憶された配列中の要素を入出力し、処理と通信を統合することで著しく高速な並列処理を実現し、著しく高速な表形式データのジョインが可能な情報処理システムおよび情報処理方法を提供することが可能となる。

【発明を実施するための最良の形態】

【0024】

[ハードウェア構成]

以下、添付図面を参照して、本発明の実施の形態につき説明を加える。図1は、本発明の実施の形態にかかる情報処理システムの概略を示すブロックダイアグラムである。図1に示すように、この実施の形態においては、複数のプロセッサ付きメモリモジュール（以下、「PMM」と称する）12-0、12-1、12-2、・・・がリング状に配置され、隣接するメモリモジュール間を、時計回りにデータを伝達する第1のバス（たとえば、符号14-0、14-1参照）、および、反時計回りにデータを伝達する第2のバス（たとえば、符号16-0、16-1参照）が接続している。第1のバスおよび第2のバスでは、PMM間のパケット通信が実行される。本実施の形態において、このパケット通信が実行される伝送路（パケット伝送路）を、第1のバスおよび第2のバスと称する。

【0025】

図2は、PMM12の構造の一例を示す図である。図2に示すように、PMM12は、命令にしたがったメモリのアクセス、演算の実行などを制御する制御回路20と、バスインタフェース(I/F)22と、メモリ24とを備えている。

【0026】

メモリ24は、複数のバンクBANK0、1、・・・、n（符号26-0、・・・、n）を有し、それぞれに、後述する所定の配列を記憶できるようになっている。

【0027】

また、制御回路20は、外部の他のコンピュータ等とのデータ授受が可能である。また、他のコンピュータが、バスアービトレーションにより、メモリの所望のバンクにアクセスできるようにしても良い。

【0028】

[データの記憶構造]

図3は、表形式データの一例を示す図である。このように、表形式のデータでは、レコードごとに種々の項目（この例では、「性別」、「年齢」、「身長」および「体重」）に値が与えられている。本実施の形態にかかる情報処理装置では、これら表形式データを、原理的には、図4に示すようなデータ形式に基づいて保持する。

【0029】

図4に示すように、順序集合の配列OrdSetには、順序番号ごとにレコード番号が値として配置される。この例では、すべてのレコードが表されるため、順序番号とレコード番号とは一致する。

【0030】

たとえば、性別に関しては、実際の項目値である「男」或いは「女」という値が、所定の順序にてソートされた値リストVLと、順序集合の配列OrdSet中の要素（レコード番号）のそれぞれに対応して、当該レコード番号が指し示す値リスト中の番号が格納された、値リストへのポインタ配列VNoとにより、表形式データを表す。この値リストVLおよびポインタ配列VNoの組み合わせを「情報ブロック」とも称する（性別に関する情報ブロックは符号401に対応する）。

【0031】

順序集合の配列OrdSet中の要素（レコード番号）が指し示す位置にある、ポインタ配列VNo中の値を特定し、さらに、その値が指し示す位置にある値リストVL中の項目値を取り出すことにより、レコード番号に対応する項目値を取得することができる。他の項目の情報ブロックについても同様の構造である。

【0032】

単一のコンピュータが、単一のメモリ（物理的には複数であっても良いが、単一のアドレス空間に配置されアクセスされるという意味で単一のメモリ）であれば、当該メモリに、順序集合の配列OrdSet、各情報ブロックを構成する値リストVLおよびポインタ配列VNoとを記憶しておけばよい。しかしながら、大量のレコードを保持するためには、その大きさにともなってメモリ容量も大きくなるため、これらを分散配置できるのが望ましい。また、処理の並列化の観点からも、分散配置された情報を分掌把握できるのが望ましい。

【0033】

そこで、本実施の形態においては、複数のPMMが、重なることなくレコードのデータを分掌把握し、PMM同士の packets 通信により、高速なジョインを実現している。

【0034】

[コンパイル処理]

まず、複数のPMMにデータを分散配置し、かつ、これらを利用可能にするための処理（コンパイル処理）について説明する。たとえば、図5に示すように、4つのPMM（PMM-0～PMM-3）に、所定のレコード数のデータを収容することを考える。この例では、レコード番号0～2に関する一連のデータ、レコード番号3、4に関する一連のデータ、レコード番号5～7に関する一連のデータ、および、レコード番号8、9に関する一連のデータを、それぞれ記憶することとした。各PMMにおいても、上記表形式データの部分は、情報ブロックの形式で記憶される。

【0035】

図6および図7は、初期的にPMM-0～4の各々にてそれぞれ分掌される表形式データの例を示す図である。これらの図から、各PMMには、項目ごとの情報ブロックの部分集合などが収容される。たとえば、図6において、項目「性別」の情報ブロック601では、もとのポインタ配列VNo（図4参照）の部分集合VNo（これも「ポインタ配列」と称する。）と、もとの値リストVL（図4参照）の部分集合VL（これも「値リスト」と称する。）とが含まれる。

【0036】

ポインタ配列VNoの要素の数は、PMMが分掌するレコードの数に一致する。これに対して、値リストVLは、ポインタ配列VNoが示す値のみが抽出される。項目「性別」につい

ては、ポインタ配列VNoの値が、値リストVL全ての要素（項目値）を指し示しているため、値リストVLと、もとの値リストVLとは一致する。その一方、項目「年齢」、「身長」および「体重」については、もとの値リストVLから、ポインタ配列中の要素が指し示す値のみが、もとの値リストVLの部分集合として取り出されることが理解できるであろう。

【0037】

さらに、分掌される情報ブロックにおいては、各PMMにおいて、ポインタ配列VNoの要素により適切に値リストVLの要素（項目値）が指し示されるように、つまり、PMM内のローカルな処理（ポインタ値の指定や項目値の指定）においても整合性が保たれるように、その要素は、対応するもとのポインタ配列VNoの要素から変換されている。

【0038】

前述したように、分掌される情報ブロックにおいては、値リストVLにおいて、当該分掌された情報ブロックにおいて必要な要素（項目値）のみを保持している。よって、ポインタ配列VNoおよび値リストVLによって、ローカルな処理の整合性は保たれる。しかしながら、PMM間での処理の整合性を保つため、各PMMにて分掌される値リストVLの要素（項目値）の、値リスト全体における位置づけ、つまり、各項目値が、値リスト全体において、所定の順序のもと何番目であるかを把握する必要がある。そこで、本実施の形態では、分掌される各情報ブロックにおいて、グローバル値番号配列GVNoを配置し、項目値に対応する値の位置を示す番号を収容できるようにしている。

【0039】

各PMMには、上記情報ブロックの部分集合を分掌するためのオフセット値(OFFSET)が割り当てられる。このオフセット値OFFSETは、PMMが分掌するレコードに関するもとの順序集合OrdSetにおける先頭の値に対応する。

【0040】

また、各PMMにおいては、ローカルな処理における整合性をたもつため、新たな順序集合OrdSetが作られる。順序集合OrdSetの要素の数は、PMMが分掌するレコード数と一致する。その一方、PMM間での処理の整合性を保つため、各PMMが分掌するレコードが、全体の中ではこういった番号（順序集合の要素）を持っているかを把握しておく必要がある。このため、全体における各レコードの番号を収容したグローバル順序集合配列GOrdを設けている。

【0041】

図8は、本実施の形態にかかるコンパイル処理を概略的に示すフローチャートである。図8に示すように、まず、各PMMに、図6～図7に示す初期的な情報ブロックが生成される（ステップ801）。これは、たとえば、外部の他のコンピュータから、PMMに、それぞれが分掌すべき、順序集合OrdSet、各情報ブロックを構成するポインタ配列VNo、値リストVL、および、オフセット値OFFSETが与えられることにより実現できる。これら配列は、各PMM内のメモリ24に記憶される。

【0042】

ステップ802以降は、各PMMにおけるローカルな処理およびPMM間のパケット通信にかかる処理に移行する。各PMMの制御回路20は、オフセット値を参照して、グローバル順序集合配列GOrd中に配置するそれぞれの値を算出し、配列中に値を配置する（ステップ802）。図9は、図6～図7に示す例でのグローバル順序集合配列GOrdへの値の配置を示す図である。ここでは、順序集合の値にオフセット値OFFSETを加えたものを、グローバル順序集合配列GOrdの対応する位置に配置すればよい。ステップ1002は、各PMMにおけるローカルな処理で実現できる。

【0043】

次いで、グローバル値リスト番号配列GVNoの値が決定される（ステップ803）。このグローバル値リスト番号配列GVNoの値の決定について、以下に詳細に説明する。以下、時計回りのバスを利用して、パケットを転送することにより、処理が進められる。図10は、各段階において伝送されるパケットの状態を示す図、図11は、各PMMにおいて、一時的に記憶されるリストを示す図である。

【0044】

図10に示すように、本実施の形態においては、ステップ1において、各PMMは、自己のVLの値をパケット化して送信する。ここでは、PMM-0～PMM-3において、それぞれ、[18, 21, 24]、[16, 28]、[16, 20, 33]および[18, 24]が時計回りに隣接するパケットに送信される。各PMMは、受信したパケット中の値と、自己のVLの値とを比較して、同一値を消去して、受信したパケットの値から同一値の消去された値からなるパケットを、さらに、時計回りに転送する。ステップ2は、最初の同一値消去が終了してパケットが送信された状態を示す。たとえば、PMM-1において、受信したパケットの値[18, 24]と自己のVLの値[18, 21, 24]とが比較される。ここでは、受信したパケットの値全てが重複しているため、PMM-1から送信されるパケットは[ϕ]となる。他のPMMにおいても同様な処理が実行されて、パケットが送信される。このような処理を繰り返すことにより、全てのPMMのVLの値が、重複値が消去されつつ、他の全てのPMMに受信されることで、図11に示すように、各PMMには、同一値の消去されたVLの値が蓄積される。この例では、4つのPMMが存在するため、4回のデータ転送が行われれば（図10のステップ1～ステップ4）、各PMMのVLの値が他のPMMに受信される。

【0045】

図11を参照すると、各PMMにて蓄積されたVLの値が一致することが理解できるであろう。次いで、受信したそれぞれのVLの値と、自己のVLの値とを比較し、他のPMMの順位を考慮した自己のVLの値の順位を決定する。それぞれのVLの値を考慮して得られた、自己のVLの値の順位の加算値の総和が、当該VLの値の、他の全てのPMMを考慮したグローバルな順位となる。図12において、重ね合わせの結果におけるGVNoの値が、各PMMのVLの対応する値の順位に相当する。

【0046】

[内部ジョイン処理]

次に、本発明にかかる情報処理装置によるジョイン処理について説明する。図13(a)は、本実施の形態にかかるジョイン処理にて結合される一方の表の論理的な構成を示す図である。このような表について、本発明に従うと、単一のコンピュータでは、図13(b)に示すようにそれぞれの項目について、値リストVLおよび値リスト中の番号を示すためのポインタ配列VNoが設けられる。図13(b)に示す構成においては、順序集合配列OrdSetの値から、ポインタ配列VNoの値を特定し、さらに、VNoが指し示す位置の値を取り出すことにより、図13(a)に示すような表を得ることができる。

【0047】

さらに、本実施の形態では、論理的には、図14に示すようにPMM-0～PMM-4の4つのPMMに表形式データの部分集合が分掌されると考える。実際には、各PMM-0～PMM-4のメモリなどには、図15に示すようなデータが記憶される。なお、図15に示す種々の配列は、PMMのRAMなどのメモリに記憶することに限定されず、アクセスをより高速にするためレジスタに記憶しておいても良い。

【0048】

図16(a)は、ジョイン処理にて結合される他方の表の論理的な構成を示す図、図16(b)は、本発明にしたがって単一のコンピュータで、図13(a)に示す表形式データを表わすために保持される種々の配列を示す図である。本実施の形態においては、PMM-0～PMM-3において、論理的に、図17に示すように表形式データの部分集合が分掌されると考える。ここでは、各PMMにおいては、図18に示すような配列が設けられる。図15および図18に示す配列において、グローバル順序集合GOrdおよびグローバル値番号配列GVNoは、上述したコンパイル処理にて得ることができる。

【0049】

ジョインは、二つのテーブルの所定の項目の値を共通化して、結合された表形式データを作ることである。図13および図16に示す表において、テーブル1（図13）の「年齢」という項目と、テーブル2（図16）の「E年齢」という項目とを共通化することで

、図 1 9 の下側に示すようなジョインされたテーブル（ビュー）が取得される。

【0050】

以下、本実施の形態に示す複数のPMMでデータ（配列）が分掌把握されている場合におけるジョイン処理についてより詳細に説明する。ここでは、図 1 3 および図 1 6 示す表（実際には、図 1 5 および図 1 8 にて各PMMにて分掌される配列）に基づいて、項目「年齢（テーブル 1）」および項目「E年齢（テーブル 2）」をジョインする場合を考える。

【0051】

[グローバル値番号配列GVNo'の共通化処理]

図 2 0 に示すフローチャート、および、図 2 1 に示す各PMMにおける配列を示す図から理解できるように、まず、それぞれのPMMは、当該PMMが分掌するテーブル 1、テーブル 2 の部分集合を構成する配列中、項目「年齢（テーブル 1）」、項目「E年齢（テーブル 2）」のそれぞれについて、グローバル値番号配列GVNo'を生成し、それぞれのGVNo'の値を初期化する（ステップ 2 0 0 1）。図 2 1 に示すように、初期的には、各PMMにおける項目「年齢」および項目「E年齢」におけるGVNo'には、昇順の値が初期値として与えられる。

【0052】

次いで、各PMMは、所定の方にPMMが持つVLの値を含むパケットを送出する（ステップ 2 0 0 2 および図 2 2）。図 2 2 に示すように、本実施の形態では、反時計回りのバスにそれぞれパケットを送出している。各PMMにおいて、テーブル 1 に関するVLの値を含むパケットは時計回りのバスに送出され、テーブル 2 に関するVLの値を含むパケットは反時計回りのバスに送出される。無論、それぞれが逆の向きに送出されるように構成しても良い。

【0053】

また、PMM-0 およびPMM-3 においては、実際にはパケットが送出されるのではなく、後述するように、次のステップで、送出にかかるVLの項目のテーブルではない、もう一方のテーブルに関して同一値を消去するために利用される。以下の説明から明らかなように、たとえば、PMM-0 のテーブル 1 に関するVLの値を含むパケットは、以下の経路で伝送される。

【0054】

PMM-1（テーブル 1 に関する処理用）－（時計回りのバス）－PMM-2（テーブル 1 に関する処理用）－（時計回りのバス）－PMM-3（テーブル 1 に関する処理用）－（PMM内のデータ転送）－PMM-3（テーブル 2 に関する処理用）－（反時計回りのバス）－PMM-2（テーブル 2 に関する処理用）－（反時計回りのバス）－PMM-1（テーブル 2 に関する処理用）－PMM-0（テーブル 2 に関する処理用）

【0055】

各PMMは、パケットを受信すると、当該パケットに含まれる他のPMMのVLの値中、自己のVLと同一の値を消去し（ステップ 2 0 0 3）、同一値の消去後の他のPMMのVLの値と、自己のVLの値とを比較し、自己のVLの順位を決定して、当該順位を配列GVNo'に格納する（ステップ 2 0 0 4）。図 2 3 に示すように、たとえば、PMM-0 のテーブル 1 に関しては、同じPMM-0 のテーブル 2 に関するVLの値を含むパケット [1 8] が内部転送される。そこで、PMM-0 は、テーブル 1 に関するVLの値 [1 8, 2 1, 2 4] とを比較して、内部転送されたパケットから同一値「1 8」を削除する。また、同一値が削除された残りのパケットは [φ] となるため、自己のVLの順位に変更はなく、したがって、GVNoの値も変わらない。

【0056】

また、PMM-1 のテーブル 1 に関しては、PMM-0 のテーブル 1 に関するVLの値を含むパケット [1 8, 2 1, 2 4] が受信される。ここでは同一値は存在しないため、同一値の削除は行われない。その一方、PMM-1 のVLの値を、上記受信したパケットに含まれる値とから、PMM-1 のVLの順位が決定され、GVNo'の値が更新される。他のPMM

Mにおいても同様の処理が実行される。

【0057】

各PMMにおいてVLの値の順位が決定され、GVNo' に値が格納されると、同一値が消去された後の値を含むパケットが、さらに、定められた方向に送出される（ステップ2005）。

【0058】

受信したパケットにおける同一値の消去、パケットの値を参照した自己のVL中の値の順位の決定、および、同一値が消去された後のパケットの送出を繰り返すことにより、図24に示すように、他のPMMのVLの値を考慮したVLのグローバルな順位を示すグローバル値番号配列GVNo' を得ることができる。図24が、各PMMにおいてグローバル値番号配列GVNo' が得られた状態を示す図である。

【0059】

ここでは、各PMMにて分掌されたテーブル1およびテーブル2に関するVLが、それぞれのPMMを通り、テーブル1に関する順位決定およびテーブル2に関する順位決定に使われることで、テーブル1およびテーブル2の共通化を図ることができる。つまり、それぞれのテーブルにおけるGVNo' の値が同じであれば、それは同じVLの値を指すことになる。

【0060】

[スレーブテーブルのソート]

次に、テーブル2において、項目「E年齢」をキーとしたソート処理が実行される。ソートは、ソートすべき項目のグローバル値番号配列GVNo（本実施の形態ではGVNo'）を用いて、グローバル順序番号配列GOrdを再配置することに相当する。なお、例示したテーブル2はすでに、項目「E年齢」でソートされている状態であるため、ここでは、上記ソート処理は省略される。

【0061】

[スレーブテーブルのカウントアップ]

ソート処理が終了すると、図25に示すように、各PMMは、スレーブテーブルに相当するテーブル2について、各PMMは、GVNo' の値ごとに、当該PMMが分掌する部分集合において、当該値が現れる個数（出現数）をカウントする（ステップ2501）。この出現数は、各PMMが分掌するレコード（順序集合配列OrdSetの要素）が、各PMMの値リストVL中の各値が、どれだけの数の、当該PMMが分掌するレコード（順序集合配列OrdSetの要素）により指し示されているかを示す。なお、GVNo' の値ごとに出現数を算出することと、GVNoの値ごとに出現数を算出することは等価である。

【0062】

より具体的には、図26に示すように、各PMMは、テーブル2について、カウントアップ領域配列Countを作成し、配列中の各値に初期値「0」を格納しておく。

【0063】

各PMMは、順序集合配列OrdSetの値を取り出して、当該OrdSetの値が指し示すGVNoの値の位置のカウント値をカウントアップする。これを繰り返すことにより（図27および図28参照）、VLの値についての出現数を、カウント配列Countに得ることができる。

【0064】

その後、各PMMは、自己のGVNoと対応する出現数との組を含むパケットを、所定の方向のバスに送出する（ステップ2502）。図29（a）に示すように、本実施の形態においては、時計方向のバスに上記組を含むパケットが送出されている。各PMMは、パケットを受信すると、パケット中の組のうち、GVNoの値を調べ、自己のグローバル値番号配列GVNoの値のうち、当該GVNoの値と同じものが存在する場合には、当該自己のGVNoの値と対応付けられたカウント配列中のカウント値に、受信したGVNoの値と組になった出現数を加算する（ステップ2503）。次いで、受信したパケットを、上記所定の方向のバスに送出する（ステップ2504）。ステップ2503～2504を所定数繰り返すことにより、各PMMにおいて、カウント配列には、他のPMMのGVNoおよびその値の出現数を考

慮した値（グローバルな出現数）を配置することができる。本実施の形態では、図29（a）～（d）に示すように、PMMの数に対応する4回だけ、データの送出手を繰り返すことで、カウンタ配列にグローバルな出現数を得ることが可能となる。無論、後述するように、各PMMにおいて、GVNoごとの出現数を取得できることから、ジョインされた表におけるグローバルな値リストの値の番号を示す配列GVNo'の出現数を把握できることも言うまでもない。

【0065】

なお、図25の処理において、PMMは受信したパケットの内容を変更せずに所定の方向に送出している。したがって、PMMは、パケットを受信したら、これを自己のメモリやレジスタに一時的に記憶した後、パケットを所定の方向に送信し、その後に、出現数の加算の処理を実行しても良い。これにより、各PMMにおいて処理を並列化することが可能となる。

【0066】

[マスタテーブルに関する処理]

次いで、各PMMは、テーブル1について、スレーブテーブルであるテーブル2との整合を図るために、2つの配列（カウンタ配列Countおよびアグリゲーション配列Aggr）を構築する。この配列の意義については後述する。

【0067】

本実施の形態においては、テーブル2についての各PMM中の配列の情報を、すべてのPMMに送信し、それぞれのPMMが、受信したテーブル2についての配列の情報に基づいて、カウンタ配列Countおよびアグリゲーション配列Aggrを生成する。図31は、PMM-0に対して、PMM-0～PMM-3から、テーブル2に関するグローバル値番号配列GVNo'の値および対応するカウンタ配列の値の組（GVNoの値、Countの値）から構成されるパケットが送信されている状態を示す。ここで、PMM-0のテーブル2に関する値の組から構成されるパケットは、バスを経ることなく、直接自身に伝達され得る。他のPMM-1～3のテーブル2に関する値の組から構成されるパケットは、バスを経てPMM-0に受信される。

【0068】

ここで、それぞれのPMMが別のバスを利用してPMM-0にデータを送信すれば、PMM-0において並列的にカウンタ配列およびアグリゲーション配列の生成が可能となる。無論、同一或いは異なるバスを利用して、PMM-1～2が、順次PMM-0にパケットを送信するように構成しても良い。

【0069】

図30は、パケットを受信したPMMにて実行される処理を示すフローチャートである。図30に示すように、PMMは、パケットを受信すると（ステップ3001）、当該パケット中のGVNo'の値を参照して、テーブル1に関するGVNo'に同じ値が存在するかを判断する（ステップ3002）。ステップ3002でイエス(Yes)と判断されると、PMMは、テーブル1のカウンタ配列Count中、対応する位置の値を、パケット中のCountの値だけカウンタアップする（ステップ3003）とともに、テーブル1のアグリゲーション配列Aggr中、上記位置の次の位置（つまり、配列の位置を示す番号が一つ増大したような位置）の値を、パケット中のCountの値だけカウンタアップする（ステップ3004）。

【0070】

その一方、ステップ3002でノー(No)と判断されると、PMMは、テーブル1のGVNo'中、パケット中のGVNo'の値より大きくかつ最小の値を見出す（ステップ3005）。次いで、PMMは、テーブル1のカウンタ配列Count中、見出された値に対応する位置の値を、パケット中のCountの値だけカウンタアップする（ステップ3006）。このような処理を、受理した全てのパケットの値について実行する（ステップ3007、3008参照）。

【0071】

図31の例において、PMM-0からテーブル2の（GVNo'の値、Countの値）として、

(1, 2) が伝達される。テーブル 1 に関して、GVNo' 中には値「1」が存在する。したがって、対応する位置のカウント配列Count中の値が、パケット中のCountの値にしたがって、「0」から「2」にカウントアップされる。また、アグリゲーション配列Aggr中、次の位置の値が、パケット中のCountの値にしたがって、「0」から「2」にカウントアップされる。

【0072】

PMM-1 からテーブル 2 の (GVNo' の値, Countの値) として、(2, 1) および (4, 2) が与えられる。この場合には、テーブル 1 に関して、GVNo' の値「2」は存在しない。そこで、テーブル 1 に関するGVNo' の値中、「2」より大きい値のうち最小の値が「3」であること（および、その値の位置が「1」であること）がわかる。そこで、その位置のAggrの値が、パケット中のCountの値にしたがって、「0」から「1」にカウントアップされる。同様に、テーブル 1 に関して、GVNo' の値「4」は存在しない。そこで、テーブル 1 に関するGVNo' の値中、「4」より大きい値のうち最小の値が「5」であること（および、その値の位置が「2」であること）がわかる。そこで、その位置のAggrの値が、パケット中のCountの値にしたがって、「0」から「2」にカウントアップされる。

【0073】

他のPMM (PMM-2 およびPMM-3) から受信した (GVNo' の値, Countの値) についても同様の処理が実行され、それぞれ、カウント配列Countおよびアグリゲーション配列の値が算出される。

【0074】

それぞれのPMMから受信したパケットに基づいてカウント配列Countおよびアグリゲーション配列Aggrを生成すると、これらを合成して最終的なカウント配列Countおよびアグリゲーション配列Aggrを生成する。ここでは、図32に示すように、PMMは、生成されたそれぞれのカウント配列Countでカウントアップされた値の総和を算出し（ステップ3201）、最終的なカウント配列Countの値とする（ステップ3202および図33（a）参照）。同様に、PMMは、生成されたアグリゲーション配列Aggrでカウントアップされた値の総和を算出し（ステップ3203および図33（a）参照）、さらに、取得した値を先頭から累算する（ステップ3204）ことで、最終的なアグリゲーション配列Aggrの値とする（ステップ3205）。図33（b）は、PMM-0において最終的に得られたカウント配列Countおよびアグリゲーション配列Aggrを示す図である。

【0075】

他のPMMにおいても同様の処理が実行されて、カウント配列Countおよびアグリゲーション配列Aggrを取得することができる。図34は、PMM-1について、PMM-0～PMM-2からの (GVNo' の値, Countの値) に基づいて、それぞれ、カウント配列Countおよびアグリゲーション配列が生成されることを説明する図、図35は、最終的なカウント配列Countおよびアグリゲーション配列Aggrの生成を説明する図である。また、図36、図38は、PMM-2、PMM-3について、PMM-0～PMM-2からの (GVNo' の値, Countの値) に基づいて、それぞれ、カウント配列Countおよびアグリゲーション配列が生成されることを、それぞれ説明する図、図37、図39は、PMM-2およびPMM-3について、最終的なカウント配列Countおよびアグリゲーション配列Aggrの生成を、それぞれ説明する図である。

【0076】

図40は、上述した処理を実行した後のPMM-0～PMM-3におけるテーブル1に関するデータを示す図である。これを、論理的な表形式に直すと、図41に示すようなものとなる。また、図42は、上述した処理を事項した後のPMM-0～PMM-3におけるテーブル2に関するデータを示す図である。これを論理的な表形式に直すと、図43に示すようになる。

【0077】

[配列CountおよびAggr生成のための他の手法]

上述した例では、各PMMにおいて、スレーブテーブルをカウントアップした後に（図

26～図28参照)、自己のGVNoと対応する出現数との組を含むパケットを、所定の方向のバスに送出している(図25のステップ2502および図29(a)参照)。ここでは、受信したパケット中に、自己のGVNoの値と同じものが存在する場合には、当該自己のGVNoの値と対応付けられたカウント配列中のカウント値に、受信したGVNoの値と組になった出現数を加算している。しかしながら、この処理を省略して、自己のGVNo'の値と、各PMMにおいてスレーブテーブルに関して生成された配列Countの対応する値(出現数)との組を含むパケットを、他のPMMに、それぞれ送信し、他のPMMにおいて、マスタテーブルに関する配列CountおよびAggrを生成しても良い。この処理によれば、図29に示すようなパケットの転送ステップを省略することができる。

【0078】

図85は、PMM-0に対して、PMM-0～PMM-3から、テーブル2に関する(GVNoの値、Countの値)から構成されるパケットが送信されている状態を示す図、図86は、PMM-0における最終的なカウント配列Countおよびアグリゲーション配列Aggrの取得を説明する図である。これらは、それぞれ、図31および図33に、ほぼ対応している。ここで留意すべきは、図31の例と比較して、PMM1から送信されるパケットにおいて、GVNo'の値に対応する出現数が異なることである。これは、本手法では、図29に示したような、同じGVNo'の値を有する出現数は、何れかのPMMにまとめるような処理が省略されているためである。また、同じ理由で、PMM-2について、GVNo'中の「4」という値に関して、この値と対応する出現数との組がパケットとして送信されている。

【0079】

図87は、PMM-1に対して、PMM-0～PMM-3から、テーブル2に関する(GVNoの値、Countの値)から構成されるパケットが送信されている状態を示す図、図88は、PMM-1における最終的なカウント配列Countおよびアグリゲーション配列Aggrの取得を説明する図であり、それぞれ、図34、35にほぼ対応する。

【0080】

図89は、PMM-2に対して、PMM-0～PMM-3から、テーブル2に関する(GVNoの値、Countの値)から構成されるパケットが送信されている状態を示す図、図90は、PMM-2における最終的なカウント配列Countおよびアグリゲーション配列Aggrの取得を説明する図であり、これらは、それぞれ、図36および図37にほぼ対応する。

【0081】

また、図91は、PMM-3に対して、PMM-0～PMM-3から、テーブル2に関する(GVNoの値、Countの値)から構成されるパケットが送信されている状態を示す図、図92は、PMM-3における最終的なカウント配列Countおよびアグリゲーション配列Aggrの取得を説明する図であり、これらは、それぞれ、図38および図39に対応する。

【0082】

この手法を利用しても、図86、88、90および92に示すように、マスタテーブルに関して、配列CountおよびAggrを生成することができる。

[配列の値の読み出し]

図40および図42に示す形式のデータからのデータの読み出しについて以下に説明する。図44は、読み出しの際にPMMにて実行される処理を示すフローチャートである。

【0083】

PMMは、読み出し要求レコード番号RNoを「0」に初期化する(ステップ4400)。次いで、当該RNo以下の最大数のSetAggrの値(以下、値「A」とも称する。)が存在するか否かを判断する(ステップ4401)。ステップ4401でノー(No)と判断されれば、RNoをインクリメントする(ステップ4402)。

【0084】

ステップ4401でイエス(Yes)と判断された場合には、PMMは、当該SetAggrに対応するOrdSetの値、および、OrdSetの値に特定される位置のVNoの値から、VNoの値の示す位置のCountの値(以下、値「C」とも称する。)を特定する(ステップ4403)。このCountの値により、表にした場合に、そのレコードが「C」個存在することがわかる。

【0085】

次いで、PMMは、 $RNo < [A + C]$ であるかを判断する（ステップ4404）。ステップ4404でノー(No)である場合には、ステップ4402に進む。その一方、ステップ4404でイエス(Yes)と判断された場合には、PMMは、テーブル2を参照するためのベースとして、Countの値と対応する位置のAggrの値を参照する（ステップ4405）。このAggrの値を以下、「Base」とも称する。値「Base」は、テーブル2のGordの値の基礎となる。

【0086】

以下、PMMは、テーブル2の値を特定するための処理を進める。まず、テーブル2に関するオフセット値Offsetが、「 $RNo - A$ 」により算出される（ステップ4406）。次いで、テーブル2のGordの値が、「 $Base + Offset$ 」に一致するものが特定され（ステップ4407）、その値と同じ位置のOrdSetの値が指し示す、テーブル2のVNoの値が特定される（ステップ4408）。

【0087】

ステップ4408に到達した場合には、ステップ4403において、テーブル1のOrdSetの値からVNoが特定され得るため、テーブル1のVLの値を取り出すことができる。VLの値は、共通化された項目に関する値のみならず、他の項目の値についても同様に特定され得る。また、ステップ4407およびステップ4408において、テーブル2のOrdSetの値から、テーブル2のVNoの値が特定される。したがって、テーブル2のVLの値も同様に取り出すことができる。取り出された値は、テーブル2の値として表中に配置され得る（ステップ4409）。スレーブテーブルであるテーブル2についても、VLの値は、共通化された項目に関する値のみならず、他の項目の値についても同様に特定され得る。

【0088】

図45～図50は、PMM-0～PMM-3における読み出しの状況を示す図である。また、図51(a)～(f)は、それぞれ、図45～図50の処理が終了した時点で読み出されたジョインテーブルを示す図である。

【0089】

各段階で読み出されたテーブルを合成すると、最終的には、図52に示すような結合された表（ビュー）を取得することが可能となる。

【0090】

[他項目ジョイン]

本発明では、単一の項目をキー項目としてジョインするだけでなく、複数の項目をキー項目としてジョインを実現することもできる。複数の項目をキー項目としてジョインすることを、ここでは「他項目ジョイン」と称する。たとえば、図13～図15に示すような表形式データ（および各PMMにおけるデータの分掌）、並びに、図53～図55に示すような表形式データ（および各PMMにおけるデータの分掌）の下で、項目「性別」と「E性別」とをジョインするとともに、項目「年齢」と「E年齢」とをジョインして、ジョインされたテーブルを作る場合を考える。図56は、ジョインの結果生成される表（ビュー）を示す。なお、この例では、項目「性別」および「年齢」のソート順が保持されるため、これら項目を含むテーブル（以下、「テーブル1」とも称する。）が、マスタテーブルとなり、項目「E年齢」および「E性別」を含むテーブル（以下、「テーブル2」とも称する）が、スレーブテーブルとなる。

【0091】

他項目ジョインにおいては、ジョインすべきそれぞれの項目について、共有化されたグローバル値番号配列GVNo' が生成される。より詳細には、PMMにて図20～図24を参照して説明した処理を実行すれば良い。上記例では、項目「性別」と「E性別」とにおいて、GVNo' を共有化した結果を図57に示し、項目「年齢」と「E年齢」とにおいて、GVNo' を共有化した結果を図58に示す。また、図59は、それぞれのGVNo' を共有化した後のマスタテーブルの状態を示し、図60は、それぞれのGVNo' を共有化した後のスレーブテーブルの状態を示す。

【0092】

次いで、マスターテーブルに関して、項目「性別」および「年齢」の共有化された値番号配列GVNo' が結合され、項目「性別×年齢」に関するGVNoが生成される。図61は、結合された項目に関するGVNoの生成を概略的に示すフローチャートである。図61に示すように、まず、各PMMは、結合される項目の一方について、順序集合OrdSetの値ごとに、対応するGVNo' の値を格納した中間リストを生成する（ステップ6101）。図62において、各PMMの順序集合配列OrdSetから値が取り出され、その値が示す、項目「性別」に関するポインタ配列VNoの値を経て、当該VNoの値が示す位置のGVNo' の値を特定することができる。PMMは、特定されたGVNo' の値を、項目「性別」に関する中間体リスト中、OrdSetの値の位置に対応する位置に配置する（図62の矢印参照）。このような処理をOrdSetの値のそれぞれについて実行することにより、項目「性別」に関する中間体リストを完成させることができる。

【0093】

同様に、PMMは、結合される項目の他方についても、順序集合OrdSetの値ごとに、対応するGVNo' の値を格納した中間リストを生成する（ステップ6102）。図63は、上述した例で、項目「年齢」に関する中間リストの生成を説明する図である。

【0094】

その後、各PMMにおいて、結合された項目の値リストVLが生成される（ステップ6103）。この結合された項目の値リストVLにおいては、項目の一方の一定の順序（たとえば昇順）になるように、項目の一方の中間リストの値および他方の中間リストの値の組が並べ替えられる。図64において、右側の項目「性格×年齢」のVLが、結合された項目の値リストに相当する。ここでは、項目「性格」が昇順になるように、「性格」の値および「年齢」の値の組が並べ替えられている。無論、これに応じて、ポインタ配列VNoの順序も並べ替えられる。

【0095】

このような処理の後、PMM間のコンパイル処理により、結合された項目に関するグローバル値番号配列GVNoが生成される（ステップ6104）。PMM間のコンパイル処理は、図8～図12を参照して説明したように実行すれば良い。図65は、上記例にて生成されたグローバル値番号配列GVNoを示す図である。

【0096】

PMMは、スレーブテーブルに関して、結合された項目（上記例では、「性別」および「年齢」）の共有化された値番号配列GVNo' を結合し、結合された項目（「性別×年齢」）に関するGVNoを生成する。図66は、上記例において、スレーブテーブルの項目「E性別」についての中間リストの生成を説明する図、図67は、スレーブテーブルの項目「E年齢」についての中間リストの生成を説明する図、図68は、結合された項目「E性別」×「E年齢」のPMM内コンパイルによる、結合された項目の値リストの生成を説明する図、図69は、PMM間のコンパイルによる、結合された項目のグローバル値番号配列GVNoの生成を説明する図である。

【0097】

これ以後の処理は、先に説明した単一の項目のジョインと同様である。ここでは、結合された項目について、スレーブテーブルのソート、スレーブテーブルのカウントアップ、マスターテーブルに関する処理、配列の値の読み出しを実行すれば良い。

【0098】

[外部ジョイン]

次に、外部ジョインについて説明する。外部ジョインにおいては、マッチングキーが相手方のテーブル（スレーブテーブル）に存在しない場合には、そこをブランクのレコードとして残すように処理される。いままで説明したジョイン（これを、「内部ジョイン」とも称する。）においては、マッチングキーが相手側に存在しない場合、そのレコードは削除されている。その一方、外部ジョインでは、ブランクレコードを挿入することで、自分のレコードを残しておくことができる。

【0099】

図13～図15に示すような表形式データ（および各PMMにおけるデータの分掌）の下で、外部ジョインを実行すると、図70の下欄に示すような表（ビュー）が生成される。図70の下欄から、スレーブテーブル（テーブル2）にマッチングキーが存在しない場合には、ジョインされたテーブルにおいてブランクとなっていることが理解できるであろう。

【0100】

外部ジョインにおいては、マスタテーブル（テーブル1）について、スレーブテーブルであるテーブル2との整合を図るためのカウント配列Countおよびアグリゲーション配列Aggrの生成が、内部ジョインと若干相違する。

【0101】

外部ジョインにおいては、各PMMにおいて、カウント配列Countおよびアグリゲーション配列Aggrが取得されたときに（図33, 35, 37および39参照）、Countの値が「0」である場合、当該値を「1」に変更するとともに、アグリゲーション配列Aggr中、対応する位置の値を「-1」にする。これは、マスタテーブルにおいて、マッチングキーが存在しない場合でも、ブランクレコードが表示できるように、表示領域を確保するための処理である。ここで、Aggrの値を「-1」とするのは、ありえない値を与えておくことで、ビューの作成時にブランクの領域を作るべきことを示すためである。

【0102】

図71（特に、右欄）を参照すると、図33, 35, 37および39に示す例において、外部ジョインのために、Countの値が「0」である要素について値が「1」に変更され、かつ、対応するAggrの値が「-1」にされていることが理解できるであろう。

【0103】

図72は、テーブル1（マスタテーブル）において、カウント配列Countおよびアグリゲーション配列Aggrなど必要な配列が生成された状態を示す図、図73は、テーブル2（スレーブテーブル）において、カウント配列Countおよびアグリゲーション配列Aggrなど必要な配列が生成された状態を示す図である。

【0104】

このように必要な配列が生成されると、各PMMにおいて配列中の値が読み出される。基本的には、内部ジョインの場合と同様に、図44に示す処理に基づいて値が読み出されるが、Aggrの値が「-1」の場合に異なる手順がある。図74に示すように、ステップ4405が終了した状態で、PMMは、値「base」が「-1」であるか否かを判断する（ステップ7401）。ステップ7401でノー(No)と判断されれば、ステップ4406に進む。その一方、ステップ7401でイエス(Yes)と判断されると、PMMは、値の無い状態を示す情報たとえば、記号「-」を、テーブル2の表中の対応する位置に配置し（ステップ7402）、ステップ4402に戻る。

【0105】

図75～図77は、PMM-0～PMM-3における読み出しの状況を示す図である。図75、図76の例は、内部ジョインの例と同様である（図45, 46参照）。これに対して、図77の例では、PMM-0において、「RNo (= 2)」以下のSetAggrの値 (= 2) が存在し（ステップ4401参照）、かつ、「RNo (= 2) < A + C (= 2 + 1)」が成立する（ステップ4404でイエス(Yes)）。しかしながら、Aggrの値 (Base) が「-1」であるため、テーブル2の値がない状態となる。したがって、テーブル2については、図78の「JOINテーブルのレコード = 2」に示されるものとなる。図78は、ジョインテーブルの各レコードにおけるテーブル1の値およびテーブル2の値を示す図である。図78に示すように、テーブル2について値が無い状態のものには、便宜上「-（マイナス）」が値として与えられている。これを組み合わせることで、図70の下欄に示すような表（ビュー）を得ることが可能となる。

【0106】

[検索等]

本実施の形態においては、ジョインテーブルの検索やソートも可能である。検索においては、検索のキー項目により、G0rdおよびOrdSetを絞り込み、絞り込まれた状態でジョインを実行すれば良い。図13～図15に示すような表形式データ（および各PMMにおけるデータの分掌）および図16～図18に示すような表形式データ（および各PMMにおけるデータの分掌）を考える。このような例では、キー項目（たとえば性別）で検索して、項目「性別」のGVNoの値が、「女性」に対応する「1」であるような、G0rdの値およびOrdSetの値の組のみと取り出しておく。このような状態で、たとえば、項目「年齢」および「E年齢」について、内部ジョインに関する処理を実行することにより、図80の左欄に示すような値の組を取得でき、これに基づいて、図80の右欄に示すような表（ビュー）を得ることができる。

【0107】

また、ジョインされたテーブルのソートも実現可能である。ソートにおいては、マスタテーブルにおいて、キーとなる項目で、GVNoにしたがって、G0rdの値を再配置しておけばよい。Gordの値の順序の入れ替えにしたがって対応するOrdSetの値も入れ替えられる。図13～図15に示すような表形式データ（および各PMMにおけるデータの分掌）および図16～図18に示すような表形式データ（および各PMMにおけるデータの分掌）を考える。このような例において、年齢でソートした上で、項目「年齢」および「E年齢」で内部ジョインを実行する場合について簡単に説明する。

【0108】

まず、マスタテーブルについて項目「年齢」をキーにして、ソートを実行する。図81は、ソート前のG0rdおよびOrdSet、並びに、ソート後のG0rdおよびOrdSetを説明する図である。PMMは、ソート後のGordおよびOrdSetを利用して、スレーブテーブル（テーブル2）についてCountを生成し、かつ、マスタテーブル（テーブル1）について必要な配列（Count、Aggr、SetAggr）を生成する。図82は、マスタテーブルに関する種々の配列を示す図、図83はスレーブテーブルに関する種々の配列を示す図である。なお、これらの図において、項目「年齢」、「E年齢」以外の項目に関する配列は省略されている。

【0109】

図82および図83に示す配列から、先に説明したような手順で読み出しをすることにより、図84の左欄に示すような値が取り出され、これに基づいて、図84の右欄に示すような表（ビュー）を得ることができる。

【0110】

本発明は、以上の実施の形態に限定されることなく、特許請求の範囲に記載された発明の範囲内で、種々の変更が可能であり、それらも本発明の範囲内に包含されるものであることは言うまでもない。

【0111】

前記実施の形態においては、PMMを、一方が時計回りにパケットを伝送する第1のバス（第1の伝送路）、他方が反時計回りにパケットを伝送する第2のバス（第2の伝送路）にて、リング状に接続している。このような構成により、パケット伝送の遅延時間などを均一化することができるため有利である。しかしながら、これに限定されず、バス型など他の形態の伝送路を採用しても良い。

【0112】

また、本実施の形態においては、メモリ、インタフェースおよび制御回路を有するPMMを利用しているが、これに限定されるものではなく、パーソナルコンピュータ、サーバなどを、ローカルな表形式データを分掌する情報処理ユニットとして、PMMの代わりに利用しても良い。或いは、単一のパーソナルコンピュータやサーバが、複数の情報処理ユニットを保持するような構成を採用しても良い。これらの場合でも、情報処理ユニットが、レコードの順位を示す値を受理し、グローバル順序集合配列G0rdを参照することにより、レコードを特定することができる。また、グローバル値番号配列を参照することにより、項目値を特定することも可能である。

【0113】

さらに、前記実施の形態においては、PMMがデータをパケット化して伝送路に送信しているがこれに限定されるものではなく、パケット以外の形態でデータを送信しても良いことは言うまでもない。

【0114】

また、情報処理ユニット間の伝送路も、いわゆるネットワーク型やバス型を採用しても良い。

【0115】

単一のパーソナルコンピュータに複数の情報処理ユニットを設けるような構成を採用することで、以下のように、本発明を利用することができる。たとえば、札幌支社、東京支社、福岡支社の3つの表形式データを用意し、通常は、各支社の単位で、検索、集計、ソートなどを実行する。さらに、3つの支社を統合したグローバルな表形式データを考えて、各支社の表形式データが、全体表のうちの部分表であるとみなし、グローバルな表形式データに関するジョインを実現することができる。

【0116】

無論、複数のパーソナルコンピュータをネットワークにて接続した場合にも、同様に、パーソナルコンピュータにて分掌されるローカルな表形式データに関する処理、および、グローバルな表形式データに関する処理を実現することもできる。

【図面の簡単な説明】

【0117】

【図1】 図1は、本発明の実施の形態にかかる情報処理システムの概略を示すブロックダイアグラムである。

【図2】 図2は、本発明の実施の形態にかかるPMMの構造の一例を示す図である。

【図3】 図3は、表形式データの一例を示す図である。

【図4】 図4は、本実施の形態において、表形式データを保持する構造の原理を説明するための図である。

【図5】 図5は、本実施の形態において、各PMMにて分掌把握される配列およびその値を説明する図である。

【図6】 図6は、初期的にPMM-0～4の各々にてそれぞれ分掌される表形式データの例を示す図である。

【図7】 図7は、初期的にPMM-0～4の各々にてそれぞれ分掌される表形式データの例を示す図である。

【図8】 図8は、本実施の形態にかかるコンパイル処理を概略的に示すフローチャートである。

【図9】 図9は、図6～図7に示す例でのグローバル順序集合配列G0rdへの値の配置を示す図である。

【図10】 図10は、本実施の形態において各段階において伝送されるパケットの状態を示す図である。

【図11】 図11は、本実施の形態にかかる各PMMにおいて、一時的に記憶されるリストを示す図である。

【図12】 図12は、本実施の形態において、一時的に記憶された配列の値を重ね合わせてGVNo'の値を取得する処理を説明する図である。

【図13】 図13(a)は、ジョイン処理にて結合される一方の表の論理的な構成を示す図、図13(b)は、本発明にしたがって単一のコンピュータで図13(a)の表を表わすために必要な種々の配列を示す図である。

【図14】 図14は、各PMMにて分掌される論理的な表を示す図である。

【図15】 図15は、各PMMにて分掌される実際の配列を示す図である。

【図16】 図16(a)は、ジョイン処理にて結合される他方の表の論理的な構成を示す図、図16(b)は、本発明にしたがって単一のコンピュータで図16(a)の表を表わすために必要な種々の配列を示す図である。

【図17】 図17は、各PMMにて分掌される論理的な表を示す図である。

- 【図 18】図 18 は、各 PMM にて分掌される実際の配列を示す図である。
- 【図 19】図 19 は、二つの表形式データをジョインして、ジョインされたテーブル（ビュー）が得られた状態を示す図である。
- 【図 20】図 20 は、グローバル値番号配列GVNo' の共通化処理を示すフローチャートである。
- 【図 21】図 21 は、グローバル値番号配列GVNo' の共通化処理を説明する図である。
- 【図 22】図 22 は、グローバル値番号配列GVNo' の共通化処理を説明する図である。
- 【図 23】図 23 は、グローバル値番号配列GVNo' の共通化処理を説明する図である。
- 【図 24】図 24 は、グローバル値番号配列GVNo' の共通化処理を説明する図である。
- 【図 25】図 25 は、本実施の形態にかかるスレーブテーブルのカウントアップ処理を示すフローチャートである。
- 【図 26】図 26 は、スレーブテーブルのカウントアップ処理を説明する図である。
- 【図 27】図 27 は、スレーブテーブルのカウントアップ処理を説明する図である。
- 【図 28】図 28 は、スレーブテーブルのカウントアップ処理を説明する図である。
- 【図 29】図 29 は、グローバルなカウント値を取得するためのパケット送信を説明する図である。
- 【図 30】図 30 は、パケットを受信した PMM にて実行される処理を示すフローチャートである。
- 【図 31】図 31 は、PMM-0 に対して、PMM-0 ~ PMM-3 から、テーブル 2 に関する (GVNo の値, Count の値) から構成されるパケットが送信されている状態を示す図である。
- 【図 32】図 32 は、各 PMM に関するカウント配列Countおよびアグリゲーション配列Aggrの値を合成して、最終的なカウント配列Countおよびアグリゲーション配列Aggrを取得する処理を示すフローチャートである。
- 【図 33】図 33 は、PMM-0 における最終的なカウント配列Countおよびアグリゲーション配列Aggrの取得を説明する図である。
- 【図 34】図 34 は、PMM-1 に対して、PMM-0 ~ PMM-3 から、テーブル 2 に関する (GVNo の値, Count の値) から構成されるパケットが送信されている状態を示す図である。
- 【図 35】図 35 は、PMM-1 における最終的なカウント配列Countおよびアグリゲーション配列Aggrの取得を説明する図である。
- 【図 36】図 36 は、PMM-2 に対して、PMM-0 ~ PMM-3 から、テーブル 2 に関する (GVNo の値, Count の値) から構成されるパケットが送信されている状態を示す図である。
- 【図 37】図 37 は、PMM-2 における最終的なカウント配列Countおよびアグリゲーション配列Aggrの取得を説明する図である。
- 【図 38】図 38 は、PMM-3 に対して、PMM-0 ~ PMM-3 から、テーブル 2 に関する (GVNo の値, Count の値) から構成されるパケットが送信されている状態を示す図である。
- 【図 39】図 39 は、PMM-3 における最終的なカウント配列Countおよびアグリゲーション配列Aggrの取得を説明する図である。
- 【図 40】図 40 は、ジョイン処理の結果得られたマスタテーブルに関する配列を示す図である。
- 【図 41】図 41 は、図 40 に示す配列を論理的な表形式に示す図である。
- 【図 42】図 42 は、ジョイン処理の結果得られたスレーブテーブルに関する配列を示す図である。

- 【図 4 3】図 4 3 は、図 4 0 に示す配列を論理的な表形式に示す図である。
- 【図 4 4】図 4 4 は、ジョインされた表の読み出し処理を示すフローチャートである。
- 【図 4 5】図 4 5 は、表の読み出し処理を説明する図である。
- 【図 4 6】図 4 6 は、表の読み出し処理を説明する図である。
- 【図 4 7】図 4 7 は、表の読み出し処理を説明する図である。
- 【図 4 8】図 4 8 は、表の読み出し処理を説明する図である。
- 【図 4 9】図 4 9 は、表の読み出し処理を説明する図である。
- 【図 5 0】図 5 0 は、表の読み出し処理を説明する図である。
- 【図 5 1】図 5 1 (a) ~ (e) は、それぞれ、図 4 5 ~ 図 5 0 の処理で得られたジョインされた表を示す図である。
- 【図 5 2】図 5 2 は最終的に得られる表を示す図である。
- 【図 5 3】図 5 3 は、は、ジョイン処理にて結合される一方の表の論理的な構成を示す図、および、本発明にしたがって単一のコンピュータで表を表わすために必要な種々の配列を示す図である。
- 【図 5 4】図 5 4 は、各 P MM にて分掌される論理的な表を示す図である。
- 【図 5 5】図 5 5 は、各 P MM にて分掌される実際の配列を示す図である。
- 【図 5 6】図 5 6 は、マスタテーブル (テーブル 1)、スレーブテーブル (テーブル 2)、および、ジョインにより得られるテーブルを示す図である。
- 【図 5 7】図 5 7 は、項目「性別」と「E 性別」とにおいて、GVNo' を共有化した結果を示す図である。
- 【図 5 8】図 5 8 は、項目「年齢」と「E 年齢」とにおいて、GVNo' を共有化した結果を示す図である。
- 【図 5 9】図 5 9 は、GVNo' を共有化した後のマスタテーブルの状態を示す図である。
- 【図 6 0】図 6 0 は、GVNo' を共有化した後のスレーブテーブルの状態を示す図である。
- 【図 6 1】図 6 1 は、結合された項目に関する GVNo の生成を概略的に示すフローチャートである。
- 【図 6 2】図 6 2 は、マスタテーブルにおける一方の項目の中間リストの生成を説明する図である。
- 【図 6 3】図 6 3 は、マスタテーブルにおける他方の項目の中間リストの生成を説明する図である。
- 【図 6 4】図 6 4 は、マスタテーブルにおける結合された項目に関する値リスト VL を説明する図である。
- 【図 6 5】図 6 5 は、結合された項目について生成されたグローバル値番号配列 GVNo を示す図である。
- 【図 6 6】図 6 6 は、スレーブテーブルの項目「E 性別」についての中間リストの生成を説明する図である。
- 【図 6 7】図 6 7 は、スレーブテーブルの項目「E 年齢」についての中間リストの生成を説明する図である。
- 【図 6 8】図 6 8 は、結合された項目「E 性別」×「E 年齢」の P MM 内コンパイルによる、結合された項目の値リストの生成を説明する図である。
- 【図 6 9】図 6 9 は、P MM 間のコンパイルによる、結合された項目のグローバル値番号配列 GVNo の生成を説明する図である。
- 【図 7 0】図 7 0 は、外部ジョインにより生成された表 (ビュー) を示す図である。
- 【図 7 1】図 7 1 は、外部ジョインにおける Count および Aggr を説明する図である。
- 【図 7 2】図 7 2 は、マスタテーブルにおいて、必要な配列が生成された状態を示す図である。
- 【図 7 3】図 7 3 は、スレーブテーブルにおいて、必要な配列が生成された状態を示す図である。

す図である。

【図 7 4】図 7 4 は、外部ジョインにおいて、ジョインされた表の読み出し処理の部分を示すフローチャートである。

【図 7 5】図 7 5 は、PMM-0 ~ PMM-3 における読み出しの状況を示す図である。

【図 7 6】図 7 6 は、PMM-0 ~ PMM-3 における読み出しの状況を示す図である。

【図 7 7】図 7 7 は、PMM-0 ~ PMM-3 における読み出しの状況を示す図である。

【図 7 8】図 7 8 は、外部ジョインにおいて読み出された情報を示す図である。

【図 7 9】図 7 9 は、メインテーブルにおける検索を説明する図である。

【図 8 0】図 8 0 は、検索およびジョインにより生成された表（ビュー）を説明する図である。

【図 8 1】図 8 1 は、メインテーブルにおけるソートを説明する図である。

【図 8 2】図 8 2 は、ソート後のマスタテーブルを説明する図である。

【図 8 3】図 8 3 は、ソート後のスレーブテーブルを説明する図である。

【図 8 4】図 8 4 は、ソートおよびジョインにより生成された表（ビュー）を説明する図である。

【図 8 5】図 8 5 は、他の手法に基づいて、PMM-0 に対して、PMM-0 ~ PMM-3 から、テーブル 2 に関する (GVNo の値, Count の値) から構成されるパケットが送信されている状態を示す図である。

【図 8 6】図 8 6 は、他の手法に基づいて、PMM-0 における最終的な配列 Count および Aggr の取得を説明する図である。

【図 8 7】図 8 7 は、他の手法に基づいて、PMM-1 に対して、PMM-0 ~ PMM-3 から、テーブル 2 に関する (GVNo の値, Count の値) から構成されるパケットが送信されている状態を示す図である。

【図 8 8】図 8 8 は、他の手法に基づいて、PMM-1 における最終的な配列 Count および Aggr の取得を説明する図である。

【図 8 9】図 8 9 は、他の手法に基づいて、PMM-2 に対して、PMM-0 ~ PMM-3 から、テーブル 2 に関する (GVNo の値, Count の値) から構成されるパケットが送信されている状態を示す図である。

【図 9 0】図 9 0 は、他の手法に基づいて、PMM-2 における最終的な配列 Count および Aggr の取得を説明する図である。

【図 9 1】図 9 1 は、他の手法に基づいて、PMM-3 に対して、PMM-0 ~ PMM-3 から、テーブル 2 に関する (GVNo の値, Count の値) から構成されるパケットが送信されている状態を示す図である。

【図 9 2】図 9 2 は、他の手法に基づいて、PMM-3 における最終的な配列 Count および Aggr の取得を説明する図である。

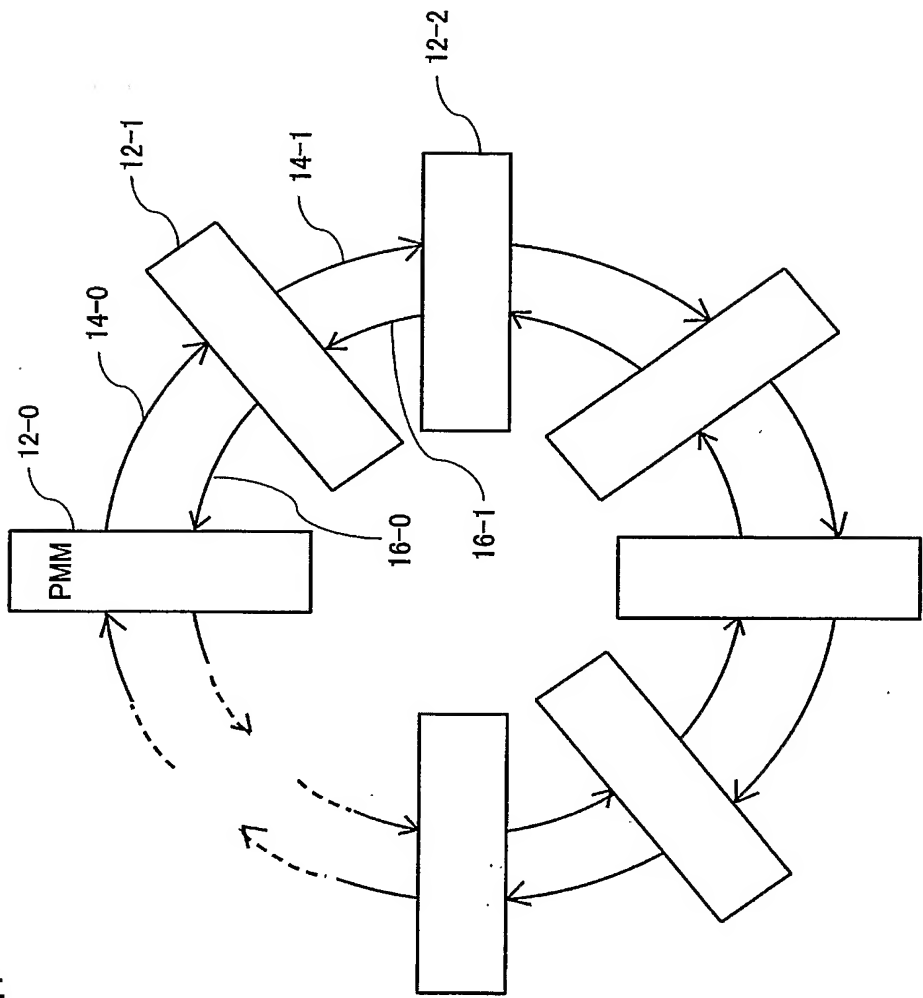
【符号の説明】

【0118】

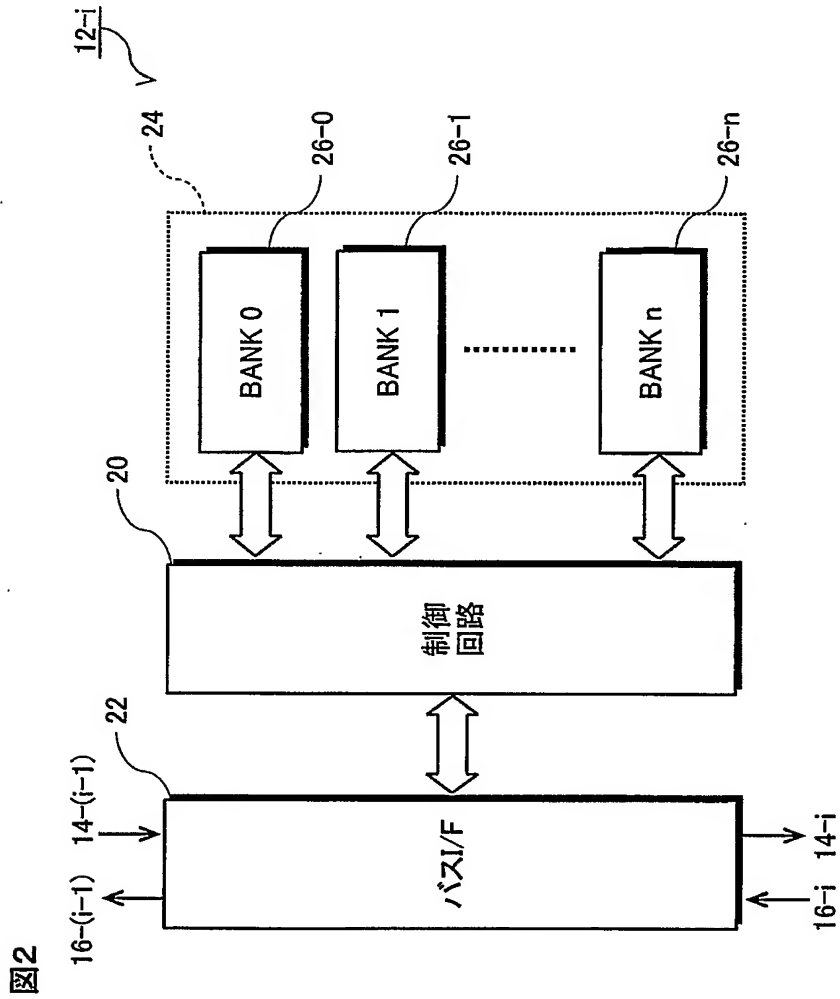
1 2	PMM
1 4	第 1 のバス
1 6	第 2 のバス
2 0	制御回路
2 2	バス I / F
2 4	メモリ
2 6	バンク

【書類名】 図面
【図 1】

図 1



【図 2】



【図 3】

性別	年齢	身長(cm)	体重(kg)
女	18	168	55
男	21	172	64
女	24	159	48
女	16	172	48
男	28	181	78
女	20	166	55
女	16	168	52
男	33	174	65
男	24	177	64
女	18	170	55

図3

【図 4】

図4

OrdSet	性別		年齢		身長		体重	
	VNo	VL	VNo	VL	VNo	VL	VNo	VL
0	1	男	0	16	0	159	0	48
1	0	女	1	18	1	166	1	52
2	1	昇順	2	20	2	168	2	55
3	1		3	21	3	170	3	64
4	0	4	24	4	172	4	65	
5	1	5	28	5	174	5	78	
6	1	6	33	昇順	6	177	昇順	
7	0	7		7	181			
8	0	8		8				
9	1	9		9				

1

【図 5】

図5

OFFSET= 0

PMM-0

	性別	年齢	身長(cm)	体重(kg)
0	女	18	168	55
1	男	21	172	64
2	女	24	159	48

OFFSET= 3

PMM-1

	性別	年齢	身長(cm)	体重(kg)
0	女	16	172	48
1	男	28	181	78

OFFSET= 5

PMM-2

	性別	年齢	身長(cm)	体重(kg)
0	女	20	166	55
1	女	16	168	52
2	男	33	174	65

OFFSET= 8

PMM-3

	性別	年齢	身長(cm)	体重(kg)
0	男	24	177	64
1	女	18	170	55

【図 6】

図6

性別 (PMM-0)			年齢 (PMM-0)			身長 (PMM-0)			体重 (PMM-0)		
VNo	VL	GVNo	VNo	VL	GVNo	VNo	VL	GVNo	VNo	VL	GVNo
0 1	0 男	0 -	0 0	0 18	0 -	0 1	0 159	0 -	0 1	0 48	0 -
1 0	1 女	1 -	1 1	1 21	1 -	1 2	1 168	1 -	1 2	1 55	1 -
2 1			2 2	2 24	2 -	2 0	2 172	2 -	2 0	2 64	2 -
昇順			昇順			昇順			昇順		

GOrd OrdSet		
0 -	0	
1 -	1	
2 -	2	
昇順		

性別 (PMM-1)			年齢 (PMM-1)			身長 (PMM-1)			体重 (PMM-1)		
VNo	VL	GVNo	VNo	VL	GVNo	VNo	VL	GVNo	VNo	VL	GVNo
0 1	0 男	0 -	0 0	0 16	0 -	0 0	0 172	0 -	0 0	0 48	0 -
1 0	1 女	1 -	1 1	1 28	1 -	1 1	1 181	1 -	1 1	1 78	1 -
昇順			昇順			昇順			昇順		

GOrd OrdSet		
0 -	0	
1 -	1	
昇順		

【図 7】

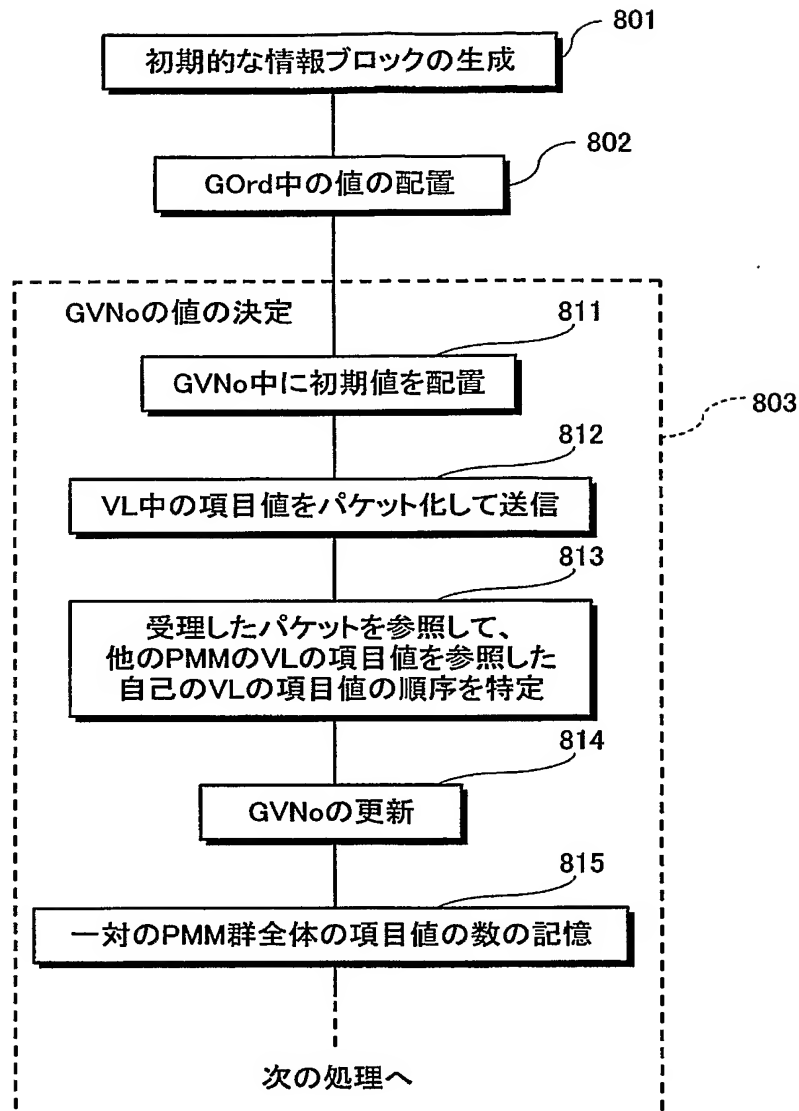
図7

性別 (PMM-2)			年齢 (PMM-2)			身長 (PMM-2)			体重 (PMM-2)		
VNo	VL	GVNo	VNo	VL	GVNo	VNo	VL	GVNo	VNo	VL	GVNo
0 1	男	0	0 1	0 16	0	0 0	0 166	0	0 1	0 52	0
1 1	女	1	1 0	1 20	1	1 1	1 168	1	1 0	1 55	1
2 0			2 2	2 33	2	2 2	2 174	2	2 2	2 65	2
昇順			昇順			昇順			昇順		

性別 (PMM-3)			年齢 (PMM-3)			身長 (PMM-3)			体重 (PMM-3)		
VNo	VL	GVNo	VNo	VL	GVNo	VNo	VL	GVNo	VNo	VL	GVNo
0 0	男	0	0 1	0 18	0	0 1	0 170	0	0 1	0 55	0
1 1	女	1	1 0	1 24	1	1 0	1 177	1	1 0	1 64	1
昇順			昇順			昇順			昇順		

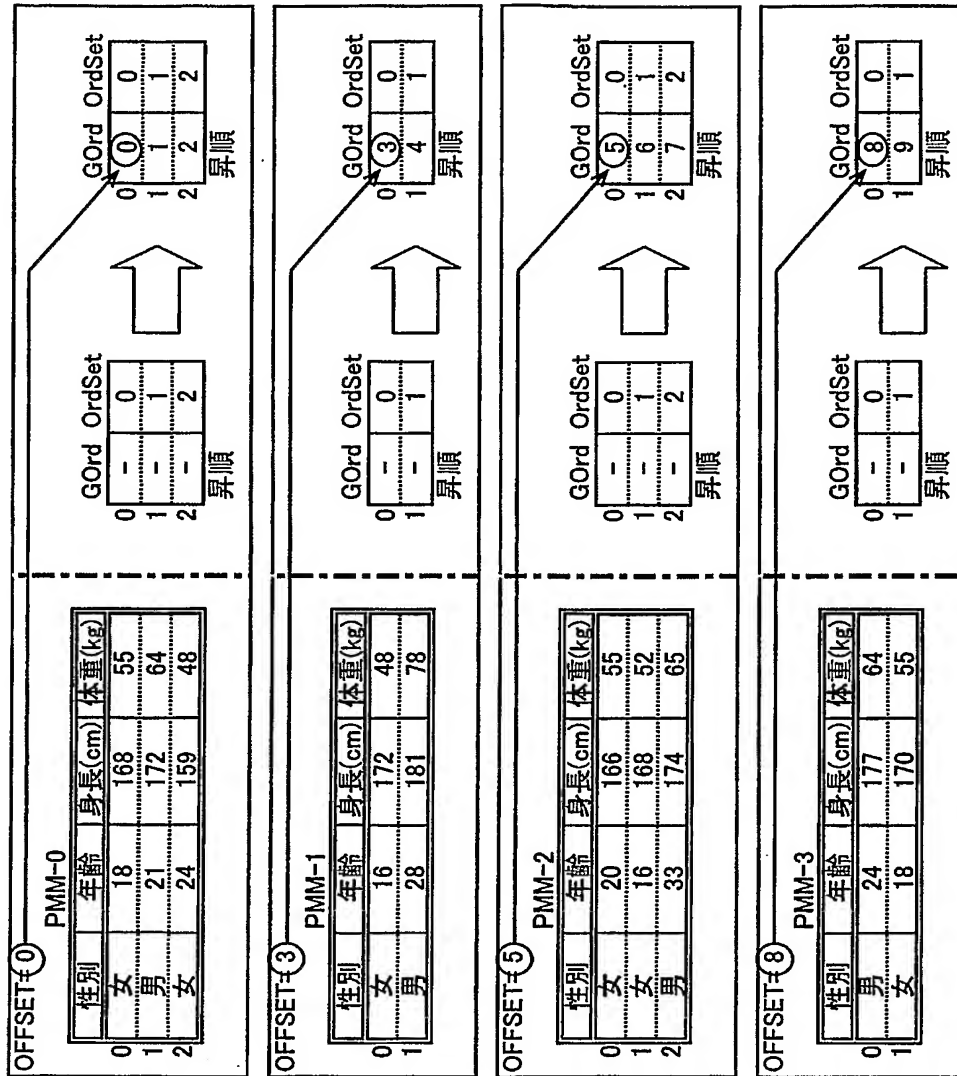
【図 8】

図8



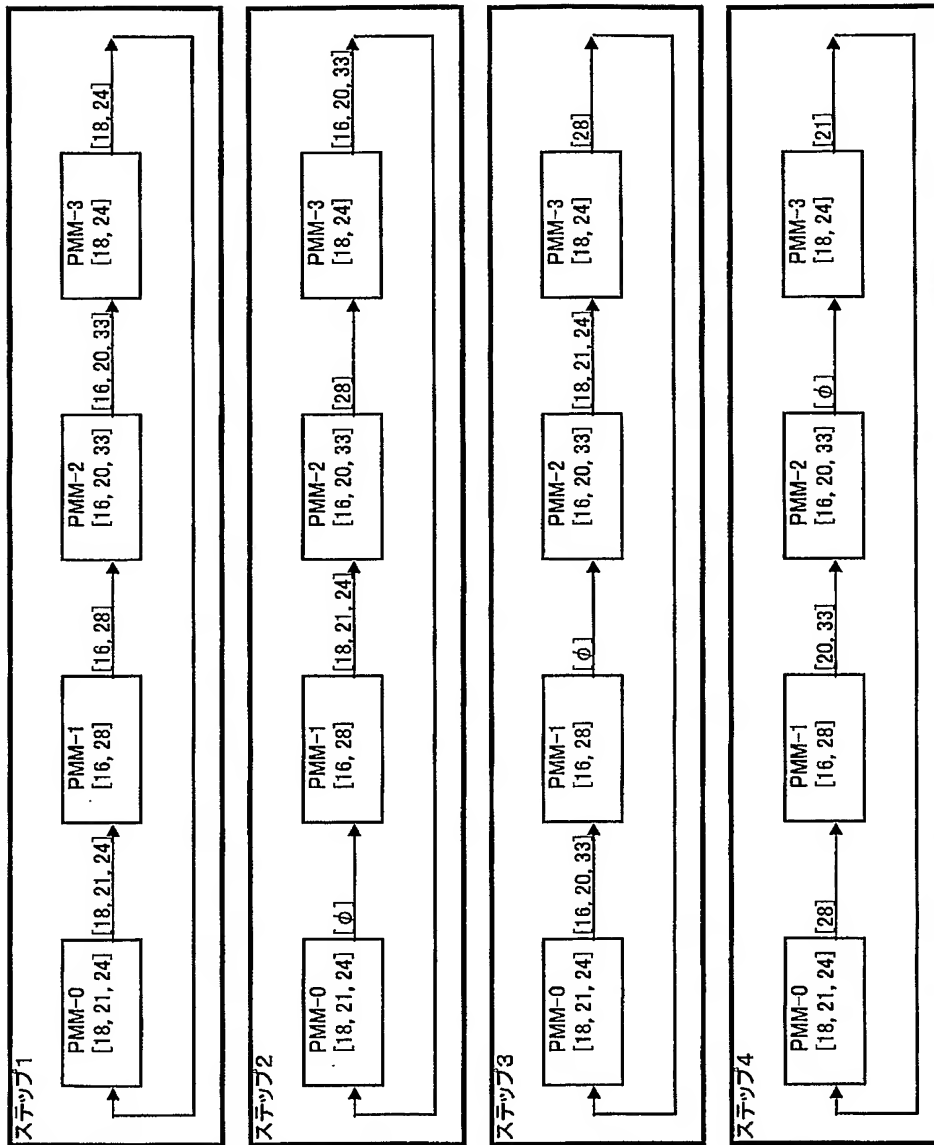
【図 9】

図9



【図 10】

図10



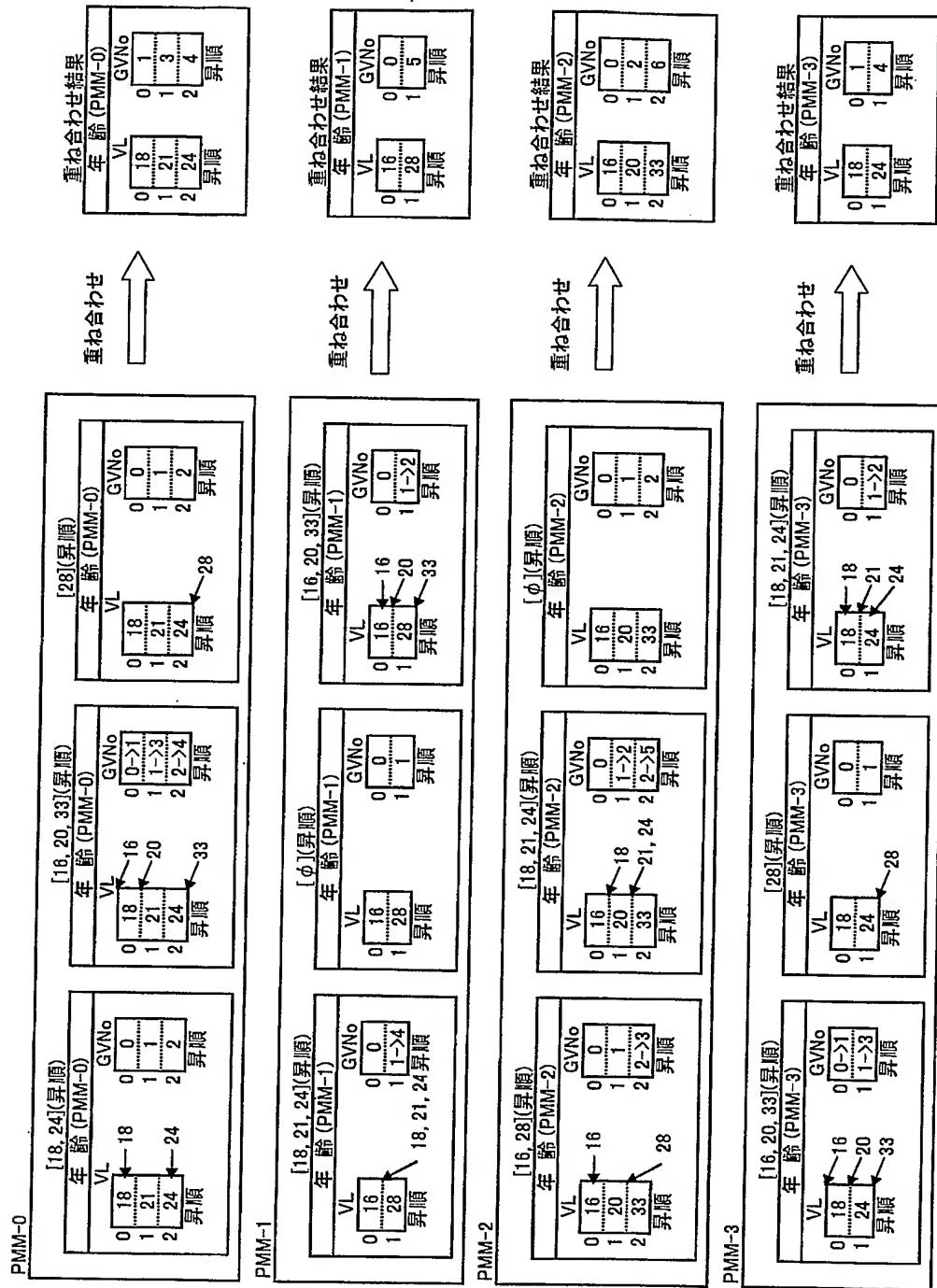
【図 11】

図 11

ステップ1			
受信済みリスト			
PMM-0 [18, 24](昇順)	PMM-1 [18, 21, 24](昇順)	PMM-2 [16, 28](昇順)	PMM-3 [16, 20, 33](昇順)
ステップ2			
受信済みリスト			
PMM-0 [18, 24](昇順) [16, 20, 33](昇順)	PMM-1 [18, 21, 24](昇順) [φ](昇順)	PMM-2 [16, 28](昇順) [18, 21, 24](昇順)	PMM-3 [16, 20, 33](昇順) [28](昇順)
ステップ3			
受信済みリスト			
PMM-0 [18, 24](昇順) [16, 20, 33](昇順) [28](昇順)	PMM-1 [18, 21, 24](昇順) [φ](昇順) [16, 20, 33](昇順)	PMM-2 [16, 28](昇順) [18, 21, 24](昇順) [φ](昇順)	PMM-3 [16, 20, 33](昇順) [28](昇順) [18, 21, 24](昇順)
ステップ4			
受信済みリスト			
PMM-0 [18, 24](昇順) [16, 20, 33](昇順) [28](昇順) [21](昇順)	PMM-1 [18, 21, 24](昇順) [φ](昇順) [16, 20, 33](昇順) [28](昇順)	PMM-2 [16, 28](昇順) [18, 21, 24](昇順) [φ](昇順) [20, 33](昇順)	PMM-3 [16, 20, 33](昇順) [28](昇順) [18, 21, 24](昇順) [φ](昇順)

【図12】

図12



【図 13】

図13

性別	年齢	身長(cm)	体重(kg)
女	18	168	55
男	21	172	64
女	24	159	48
女	16	172	48
男	28	181	78
女	20	166	55
女	16	168	52
男	33	174	65
男	24	177	64
女	18	170	55

性別	年齢	身長	体重
<div> <div>VL</div> <div>男</div> <div>女</div> <div>昇順</div> </div>	<div> <div>VL</div> <div>0 16</div> <div>1 18</div> <div>2 20</div> <div>3 21</div> <div>4 24</div> <div>5 28</div> <div>6 33</div> <div>昇順</div> </div>	<div> <div>VL</div> <div>0 159</div> <div>1 166</div> <div>2 168</div> <div>3 170</div> <div>4 172</div> <div>5 174</div> <div>6 177</div> <div>7 181</div> <div>昇順</div> </div>	<div> <div>VL</div> <div>0 48</div> <div>1 52</div> <div>2 55</div> <div>3 64</div> <div>4 65</div> <div>5 78</div> <div>昇順</div> </div>
<div> <div>VNo</div> <div>0 1</div> <div>1 0</div> <div>2 1</div> <div>3 1</div> <div>4 0</div> <div>5 1</div> <div>6 1</div> <div>7 0</div> <div>8 0</div> <div>9 1</div> </div>	<div> <div>VNo</div> <div>0 1</div> <div>1 3</div> <div>2 4</div> <div>3 0</div> <div>4 5</div> <div>5 2</div> <div>6 0</div> <div>7 6</div> <div>8 4</div> <div>9 1</div> </div>	<div> <div>VNo</div> <div>0 2</div> <div>1 4</div> <div>2 0</div> <div>3 4</div> <div>4 7</div> <div>5 1</div> <div>6 2</div> <div>7 5</div> <div>8 6</div> <div>9 3</div> </div>	<div> <div>VNo</div> <div>0 2</div> <div>1 3</div> <div>2 0</div> <div>3 0</div> <div>4 5</div> <div>5 2</div> <div>6 1</div> <div>7 4</div> <div>8 3</div> <div>9 2</div> </div>

OrdSet	0	1	2	3	4	5	6	7	8	9
--------	---	---	---	---	---	---	---	---	---	---

【図 14】

図 14

OFFSET= 0

PMM-0

	性別	年齢	身長(cm)	体重(kg)
0	女	18	168	55
1	男	21	172	64
2	女	24	159	48

OFFSET= 3

PMM-1

	性別	年齢	身長(cm)	体重(kg)
0	女	16	172	48
1	男	28	181	78

OFFSET= 5

PMM-2

	性別	年齢	身長(cm)	体重(kg)
0	女	20	166	55
1	女	16	168	52
2	男	33	174	65

OFFSET= 8

PMM-3

	性別	年齢	身長(cm)	体重(kg)
0	男	24	177	64
1	女	18	170	55

【図 15】

図15

OFFSET= 0

GOrd OrdSet
0 0 0 0
1 1 1 1
2 2 2 2
昇順

性別 (PMM-0)				年齢 (PMM-0)				身長 (PMM-0)				体重 (PMM-0)			
VNo	VL	GVNo		VNo	VL	GVNo		VNo	VL	GVNo		VNo	VL	GVNo	
0 1 1 0 2 1	男 女 昇順	0 0 1 1 2 1	昇順	0 0 1 1 2 2	0 18 1 21 2 24	0 1 1 3 2 4	昇順	0 1 1 2 2 0	0 159 1 168 2 172	0 0 1 2 2 4	昇順	0 1 1 2 2 0	0 48 1 55 2 64	0 0 1 1 2 2	昇順

OFFSET= 3

GOrd OrdSet
0 3 0 0
1 4 1 1
昇順

性別 (PMM-1)				年齢 (PMM-1)				身長 (PMM-1)				体重 (PMM-1)			
VNo	VL	GVNo		VNo	VL	GVNo		VNo	VL	GVNo		VNo	VL	GVNo	
0 1 1 0	男 女 昇順	0 0 1 1 2 1	昇順	0 0 1 1	0 16 1 28	0 0 1 5	昇順	0 0 1 1	0 172 1 181	0 4 1 7	昇順	0 0 1 1	0 48 1 78	0 0 1 1	昇順

OFFSET= 5

GOrd OrdSet
0 5 0 0
1 6 1 1
2 7 2 2
昇順

性別 (PMM-2)				年齢 (PMM-2)				身長 (PMM-2)				体重 (PMM-2)			
VNo	VL	GVNo		VNo	VL	GVNo		VNo	VL	GVNo		VNo	VL	GVNo	
0 1 1 1 2 0	男 女 昇順	0 0 1 1 2 1	昇順	0 1 1 0 2 2	0 16 1 20 2 33	0 0 1 2 2 6	昇順	0 0 1 1 2 2	0 166 1 168 2 174	0 1 1 2 2 5	昇順	0 1 1 0 2 2	0 52 1 55 2 65	0 1 1 2 2 4	昇順

OFFSET= 8

GOrd OrdSet
0 8 0 0
1 9 1 1
昇順

性別 (PMM-3)				年齢 (PMM-3)				身長 (PMM-3)				体重 (PMM-3)			
VNo	VL	GVNo		VNo	VL	GVNo		VNo	VL	GVNo		VNo	VL	GVNo	
0 0 1 1	男 女 昇順	0 0 1 1 2 1	昇順	0 1 1 0	0 18 1 24	0 1 1 4	昇順	0 1 1 0	0 170 1 177	0 3 1 6	昇順	0 1 1 0	0 55 1 64	0 2 1 3	昇順

【図 16】

図16

E年齢	イベント
18	H2
18	C1
20	A
22	C2
22	P1
27	P2
33	M

OrdSet
0
1
2
3
4
5
6

E年齢	イベント
VNo	VL
0	3
1	1
2	0
3	2
4	5
5	6
6	4
0	18
1	20
2	22
3	27
4	33
	昇順
0	A
1	C1
2	C2
3	H2
4	M
5	P1
6	P2
	昇順

【図 1 7】

図17

PMM-0 OFFSET = 0

	E年齢	イベント
0	18	H2
1	18	C1

PMM-1 OFFSET = 2

	E年齢	イベント
0	20	A
1	22	C2

PMM-2 OFFSET = 4

	E年齢	イベント
0	22	P1
1	27	P2

PMM-3 OFFSET = 6

	E年齢	イベント
0	33	M

【図 18】

図 18

OFFSET= 0

GOrd	OrdSet
0	0
1	1

昇順

E年齢(PMM-0)			イベント(PMM-0)		
VNo	VL	GVNo	VNo	VL	GVNo
0	0	0	0	C1	0
1	18	0	1	H2	1
	昇順	昇順		昇順	昇順

OFFSET= 2

GOrd	OrdSet
0	2
1	3

昇順

E年齢(PMM-1)			イベント(PMM-1)		
VNo	VL	GVNo	VNo	VL	GVNo
0	0	0	0	A	0
1	1	1	1	C2	2
	昇順	昇順		昇順	昇順

OFFSET= 4

GOrd	OrdSet
0	4
1	5

昇順

E年齢(PMM-2)			イベント(PMM-2)		
VNo	VL	GVNo	VNo	VL	GVNo
0	0	0	0	P1	0
1	1	2	1	P2	5
	昇順	昇順		昇順	昇順

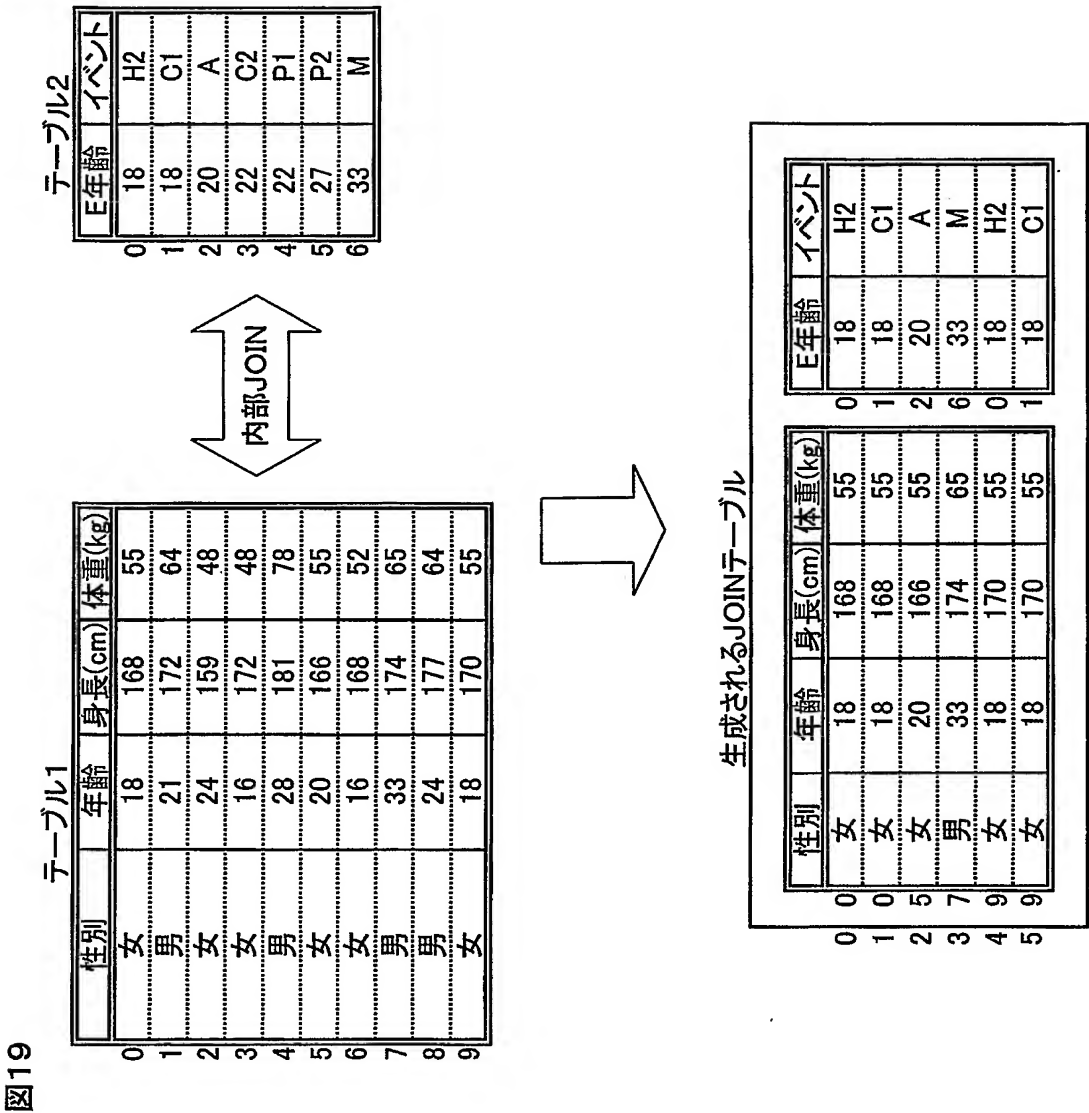
OFFSET= 6

GOrd	OrdSet
0	6
1	0

昇順

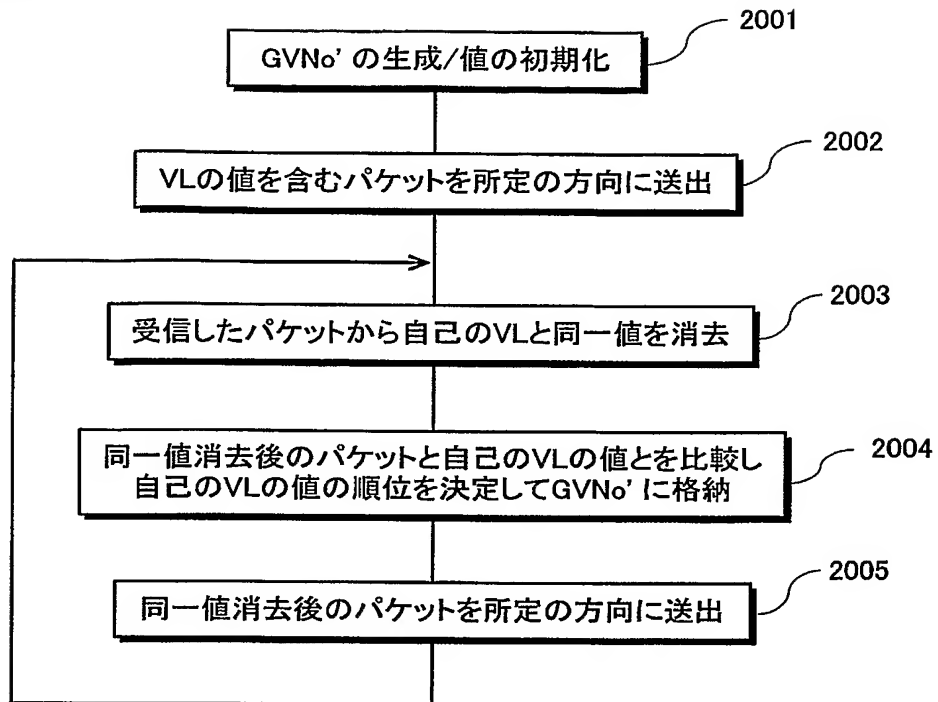
E年齢(PMM-3)			イベント(PMM-3)		
VNo	VL	GVNo	VNo	VL	GVNo
0	0	0	0	M	0
1	33	4	1		4
	昇順	昇順		昇順	昇順

【図 1 9】



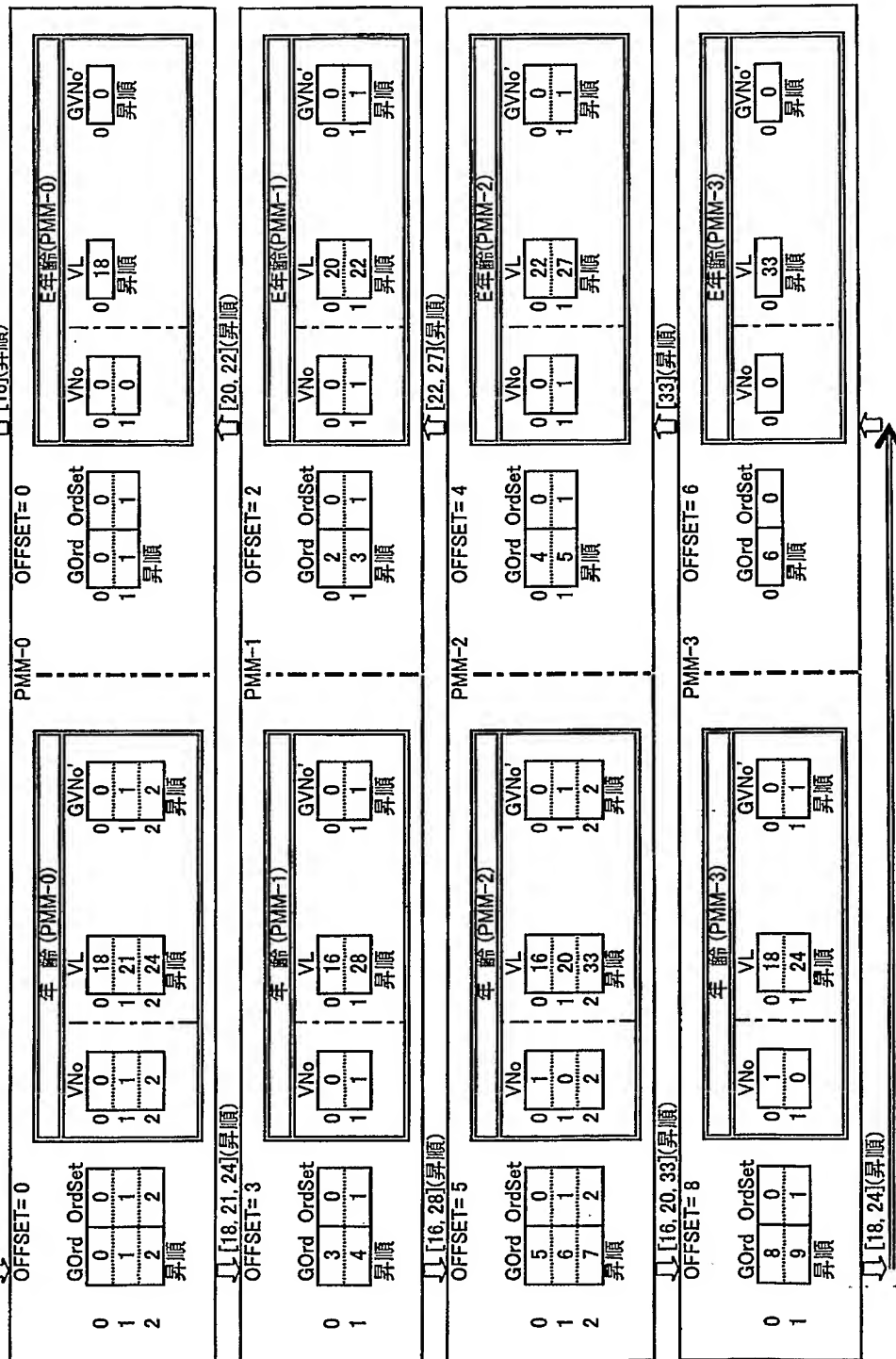
【図 20】

図20

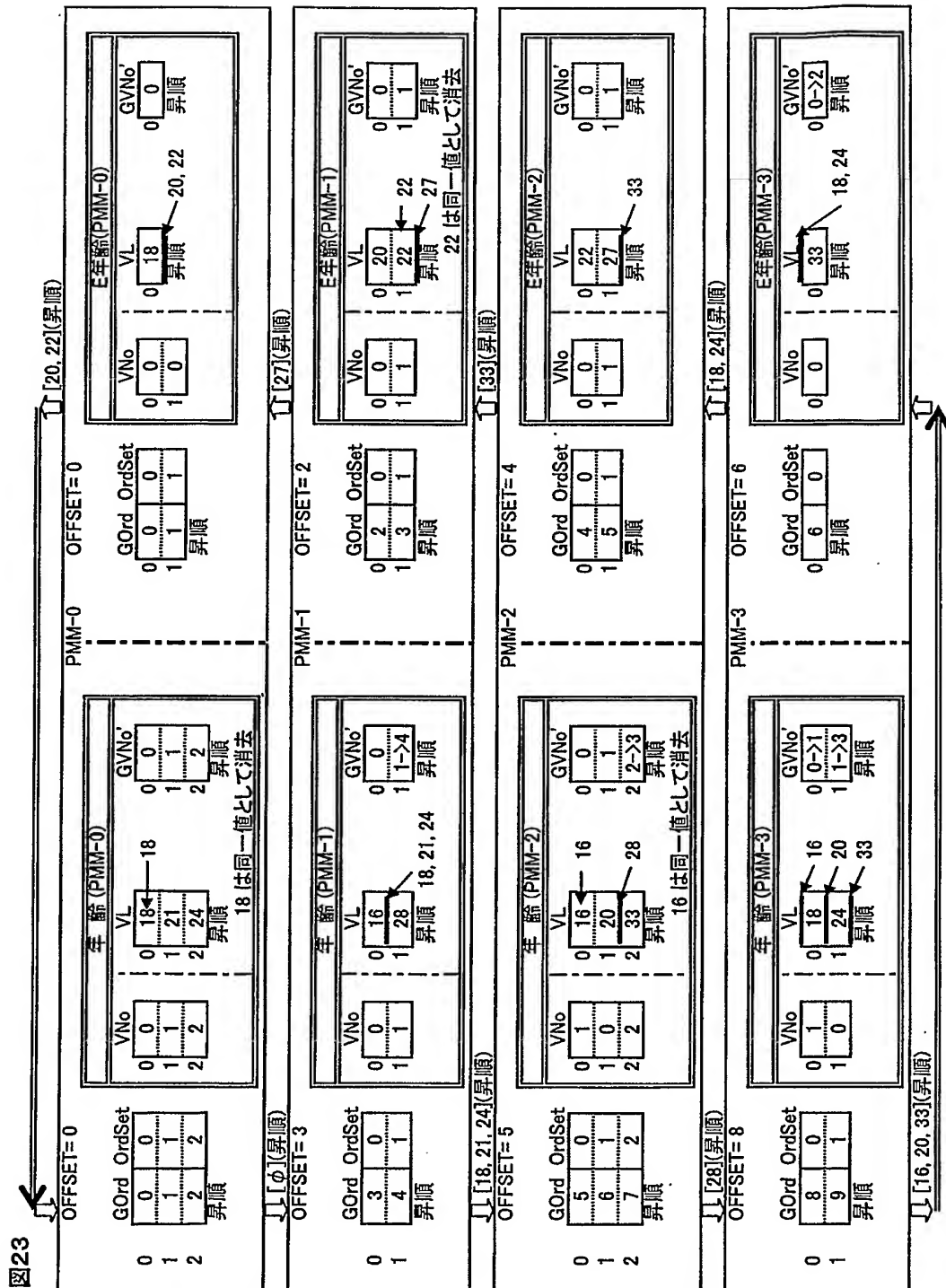


【図 22】

図22 各PMMの持つVLを送出する。

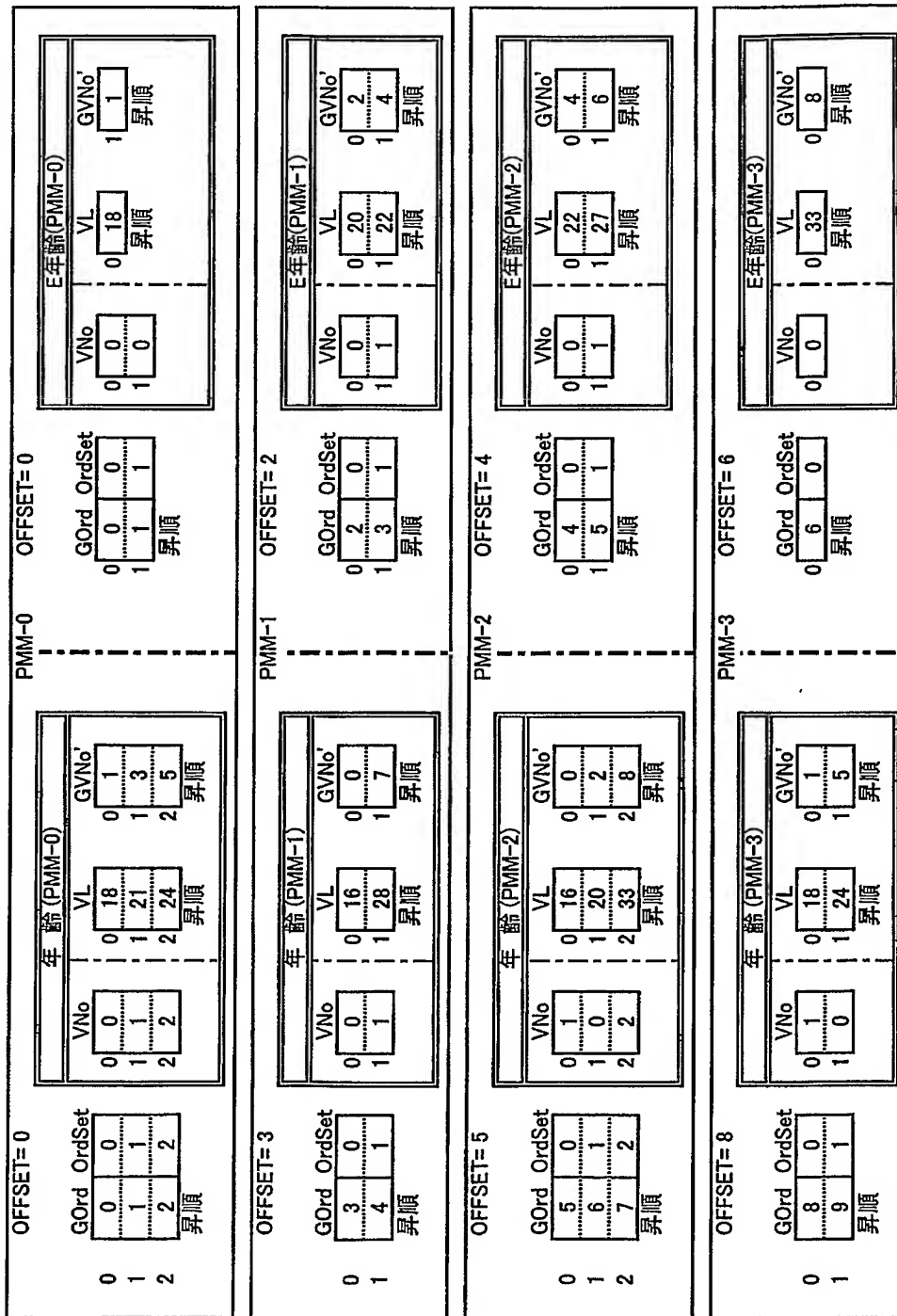


【図 23】



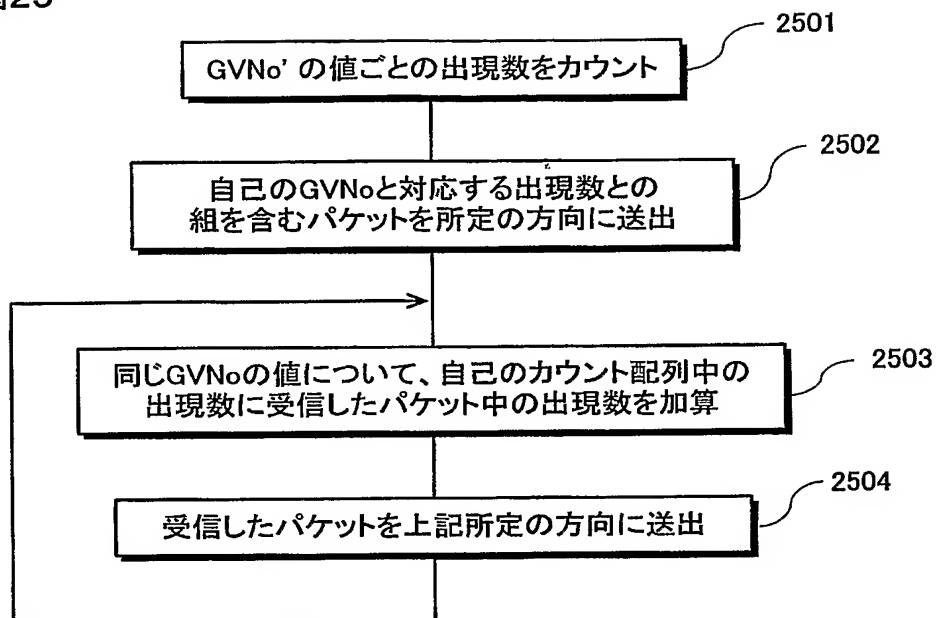
【図 24】

図24



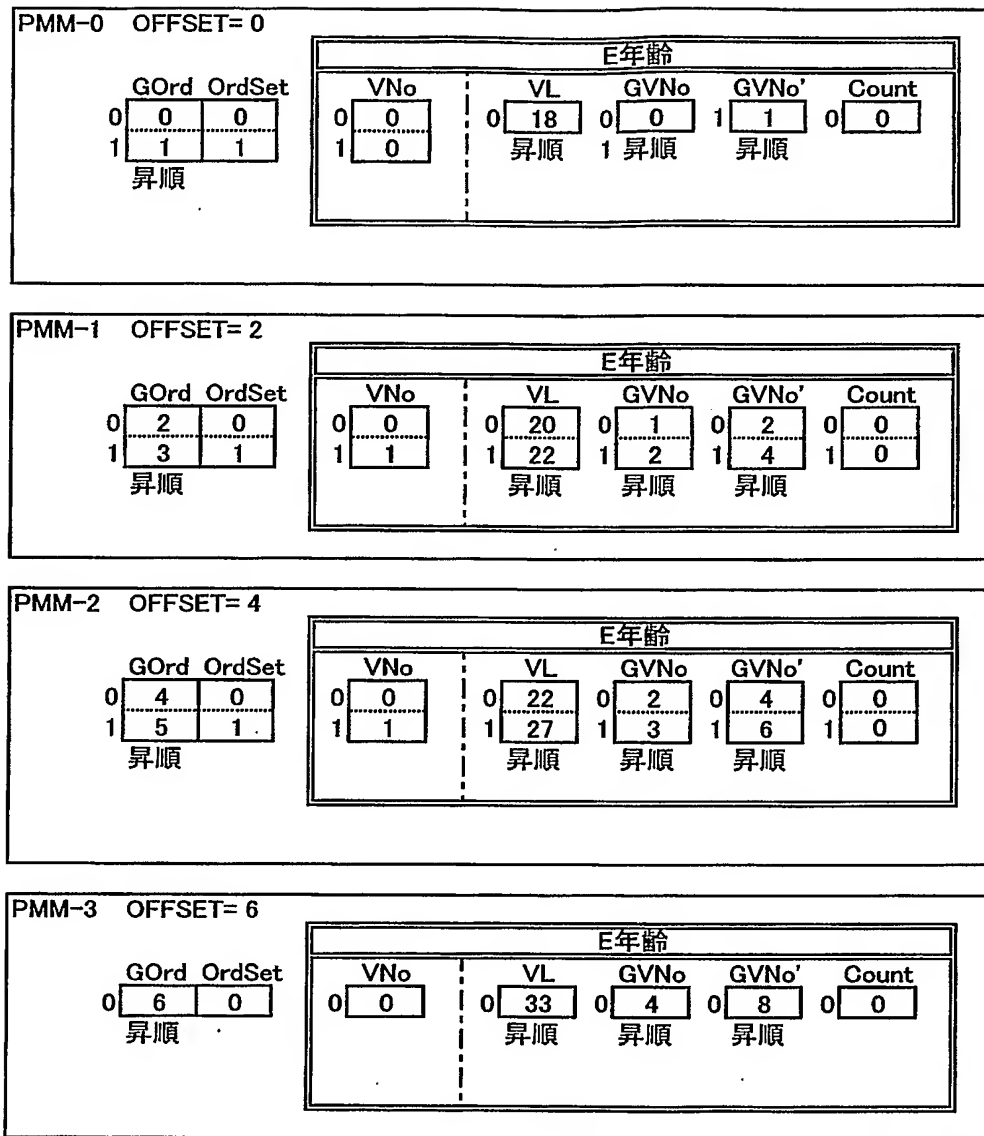
【図 25】

図25



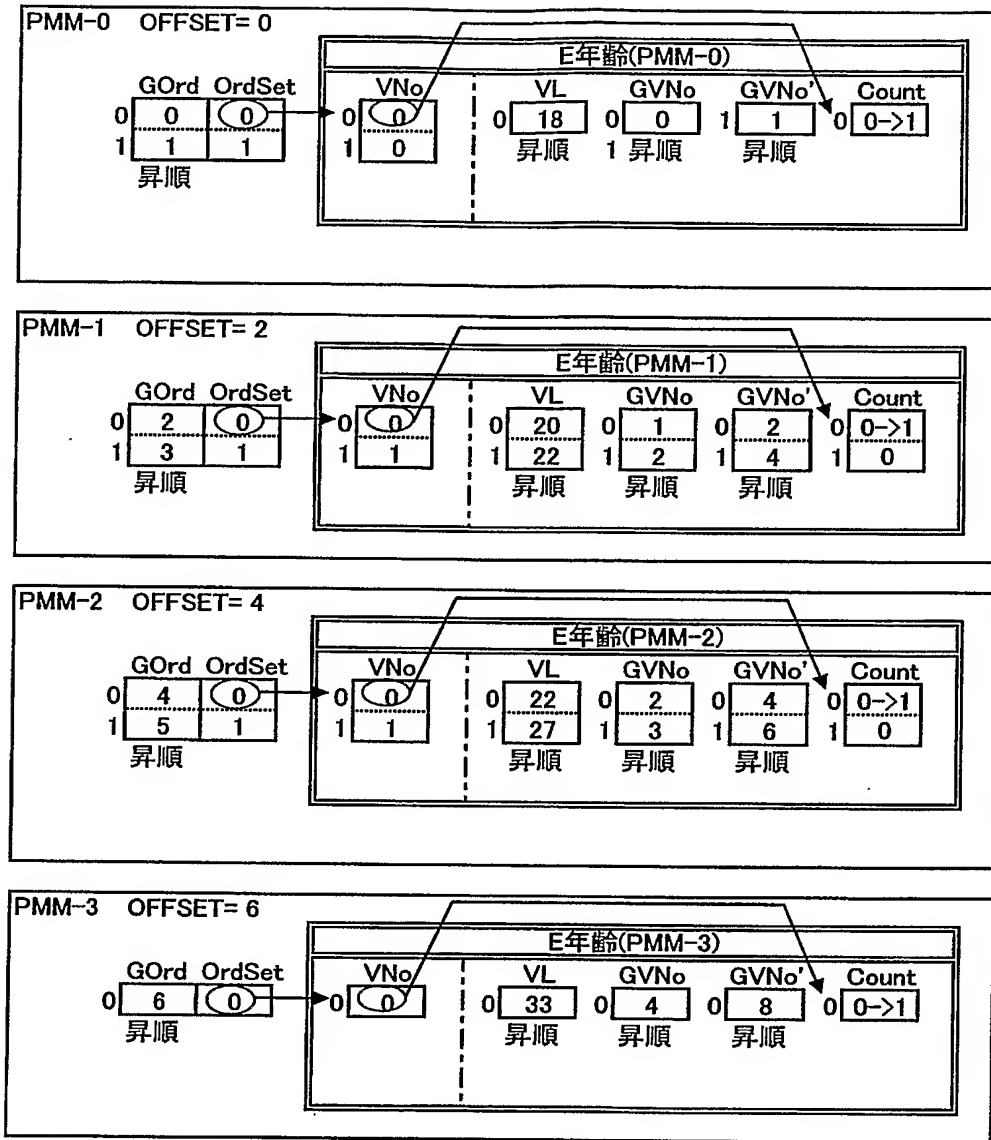
【図 26】

図26



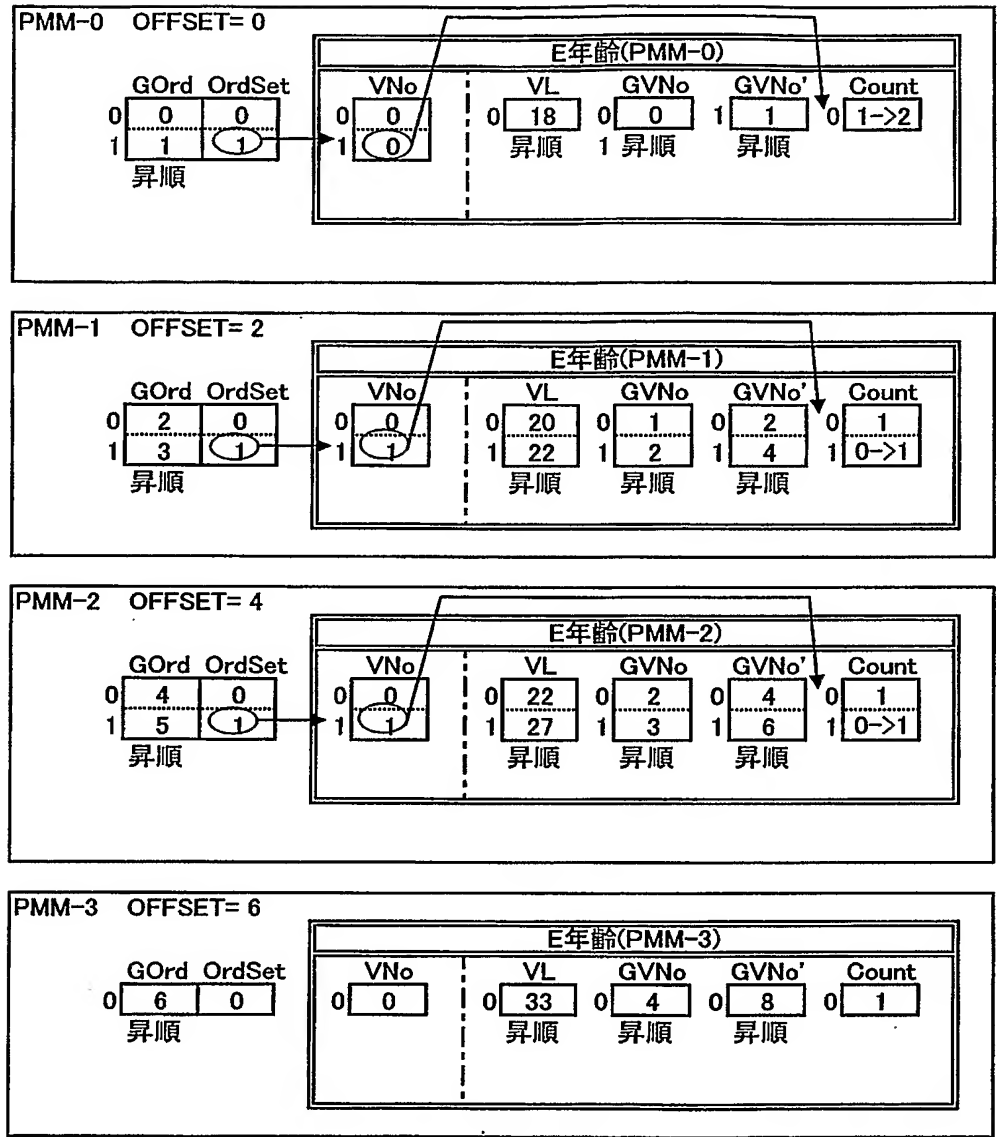
【図 27】

図27

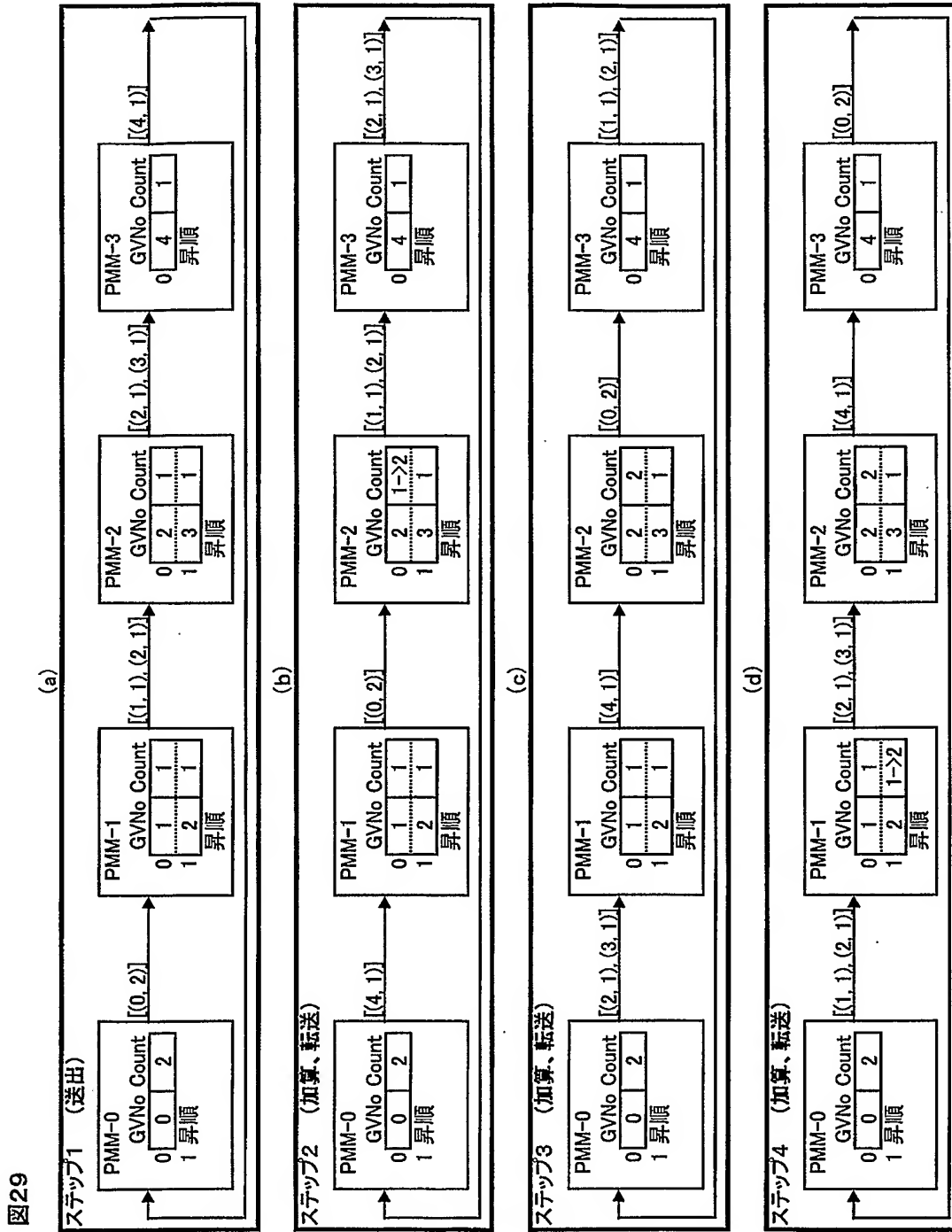


【図 28】

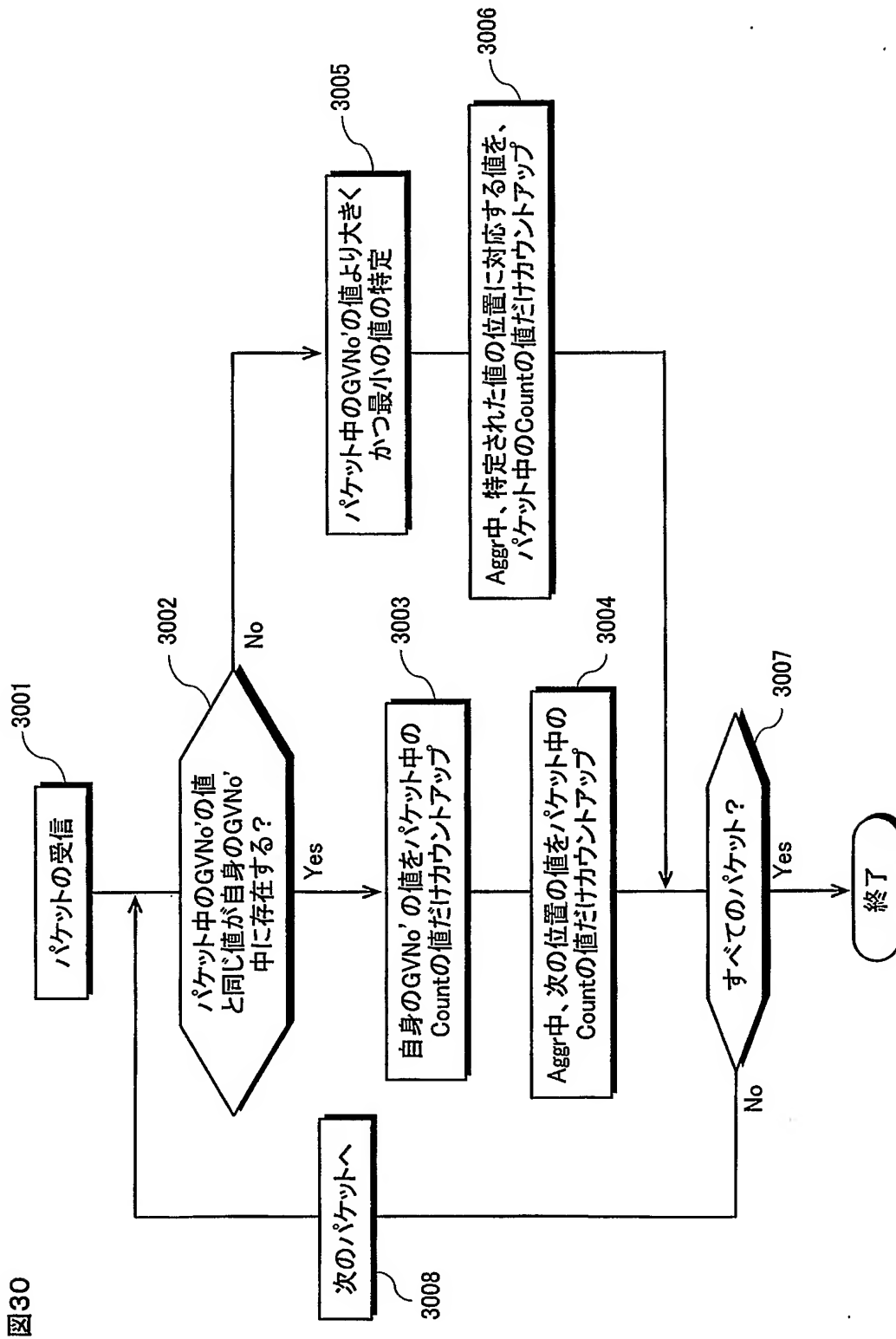
図28



【図 29】

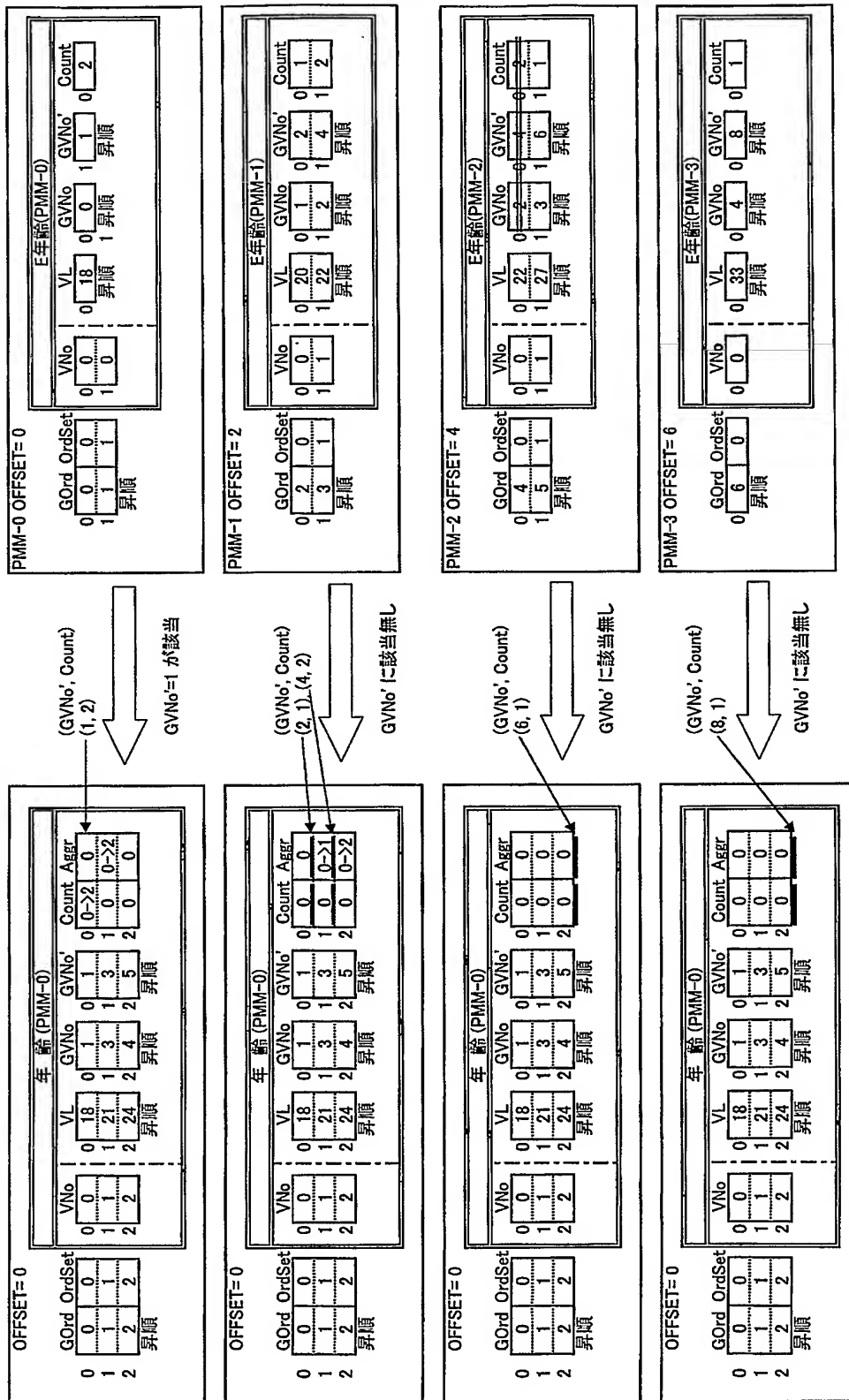


【図 30】



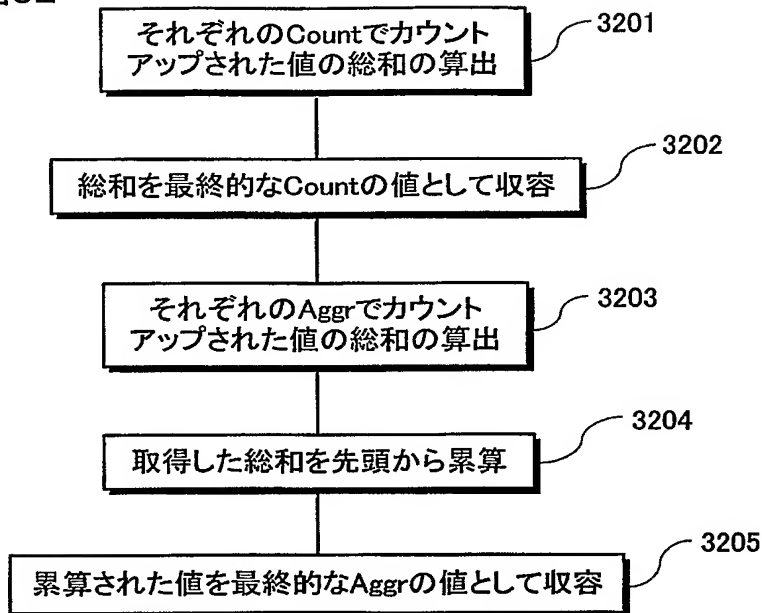
【図 31】

図31



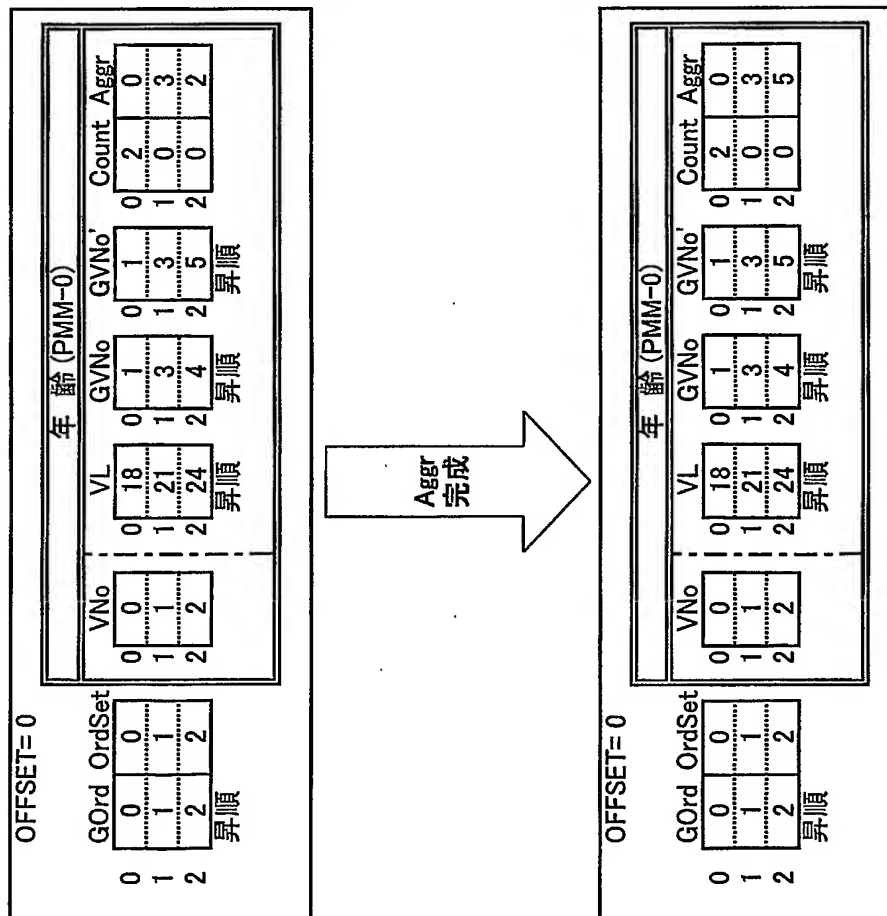
【図 3 2】

図32

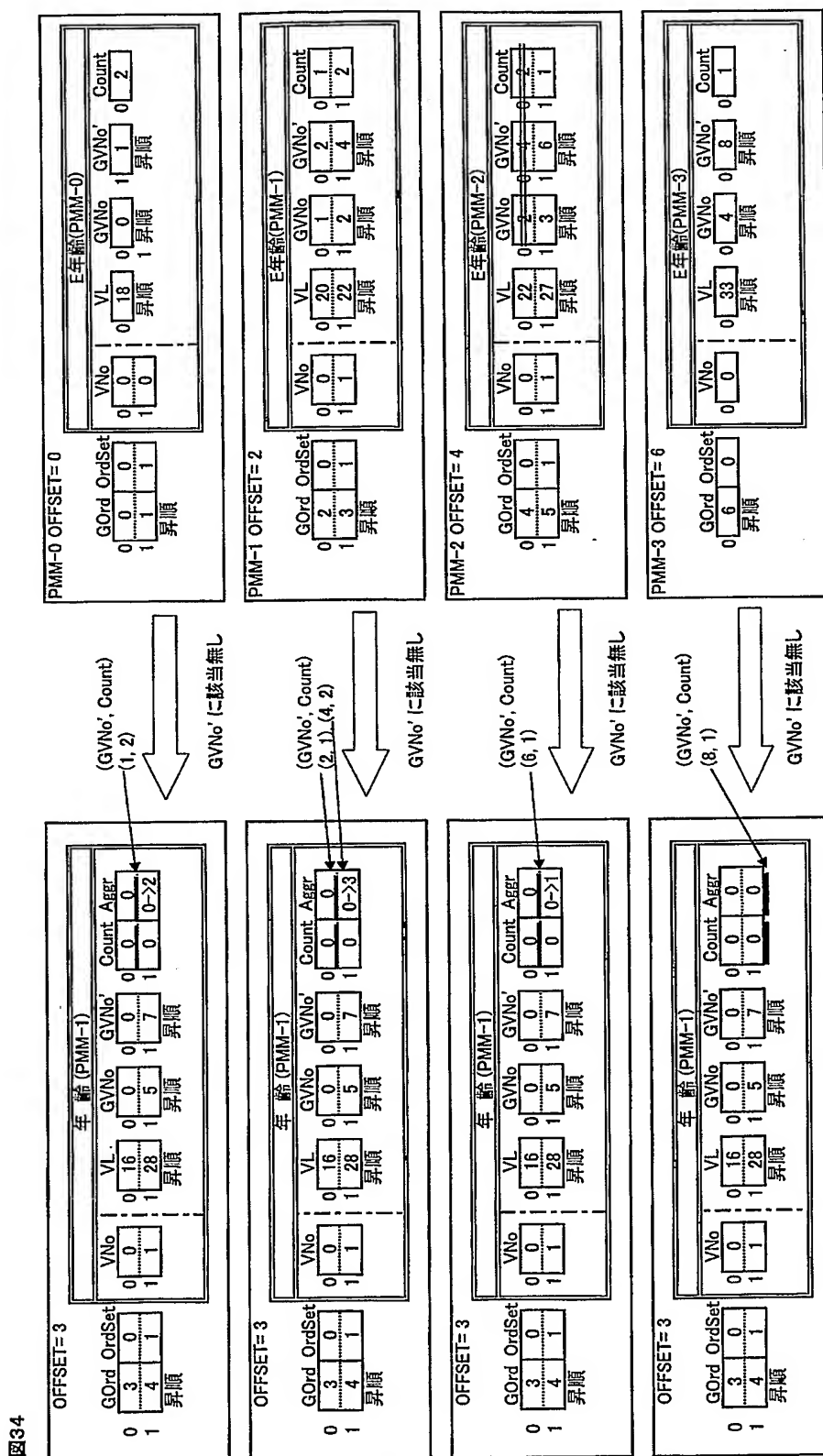


【図 3 3】

図33

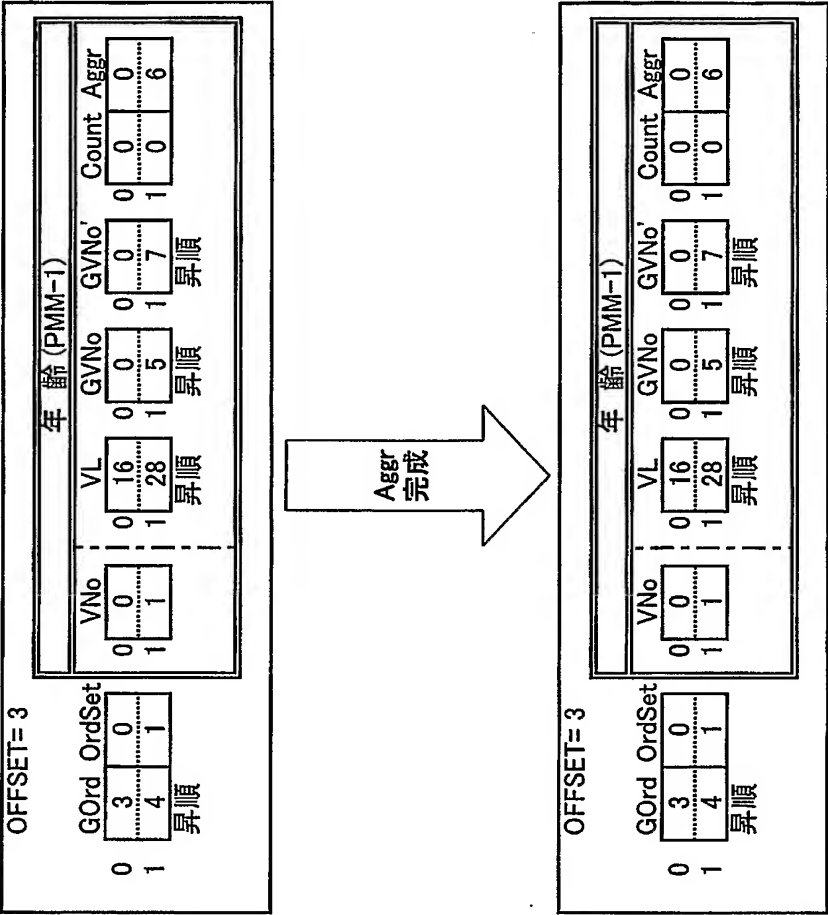


【図 3 4】



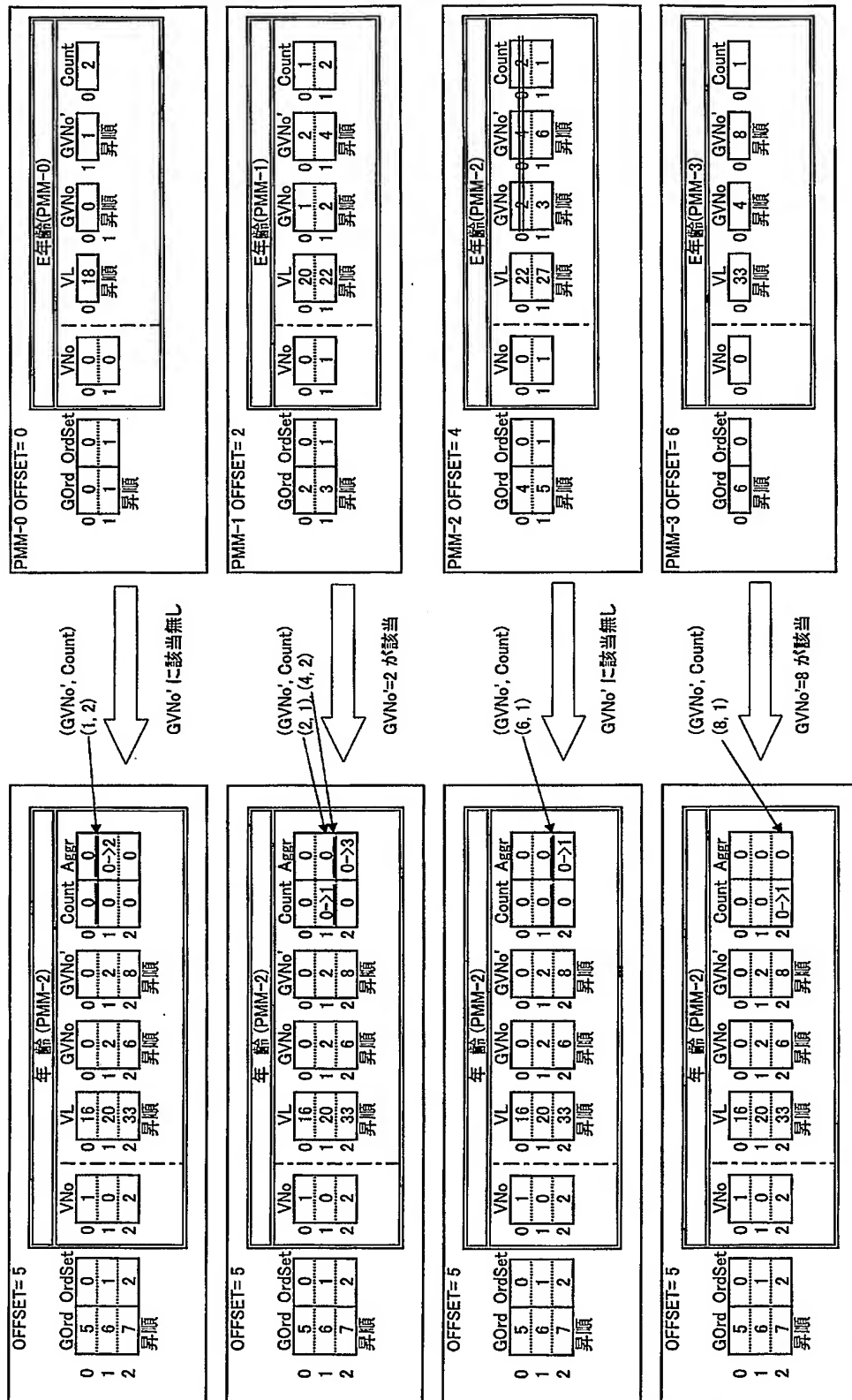
【図 35】

図35



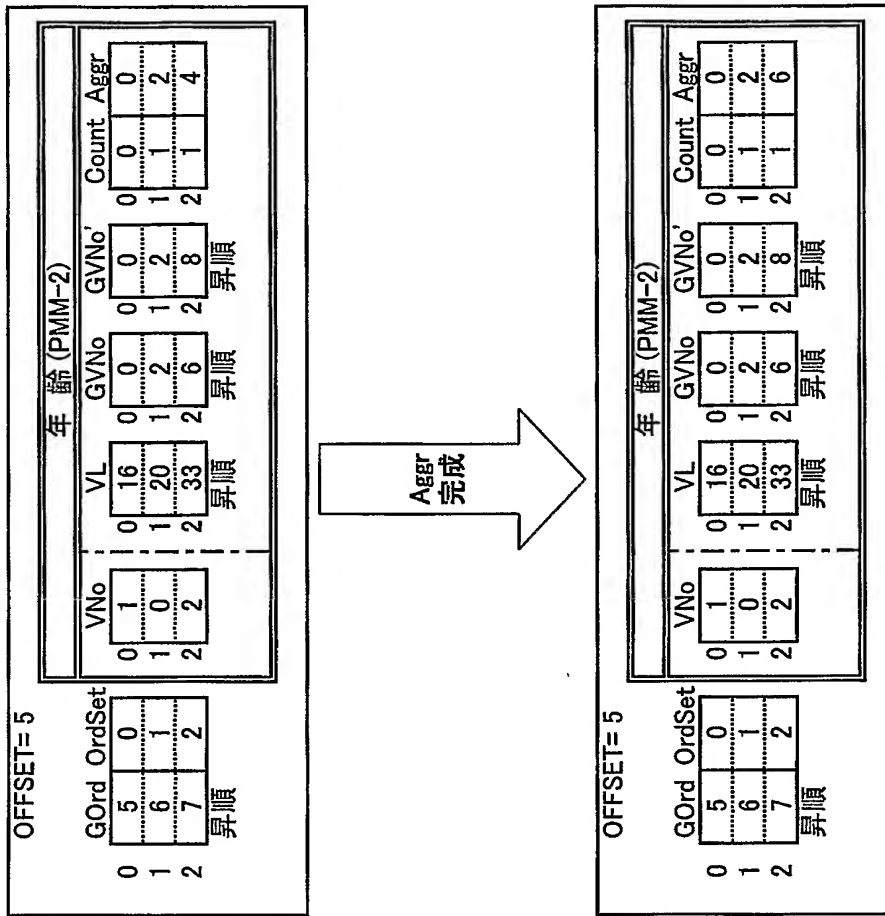
【図 36】

図36



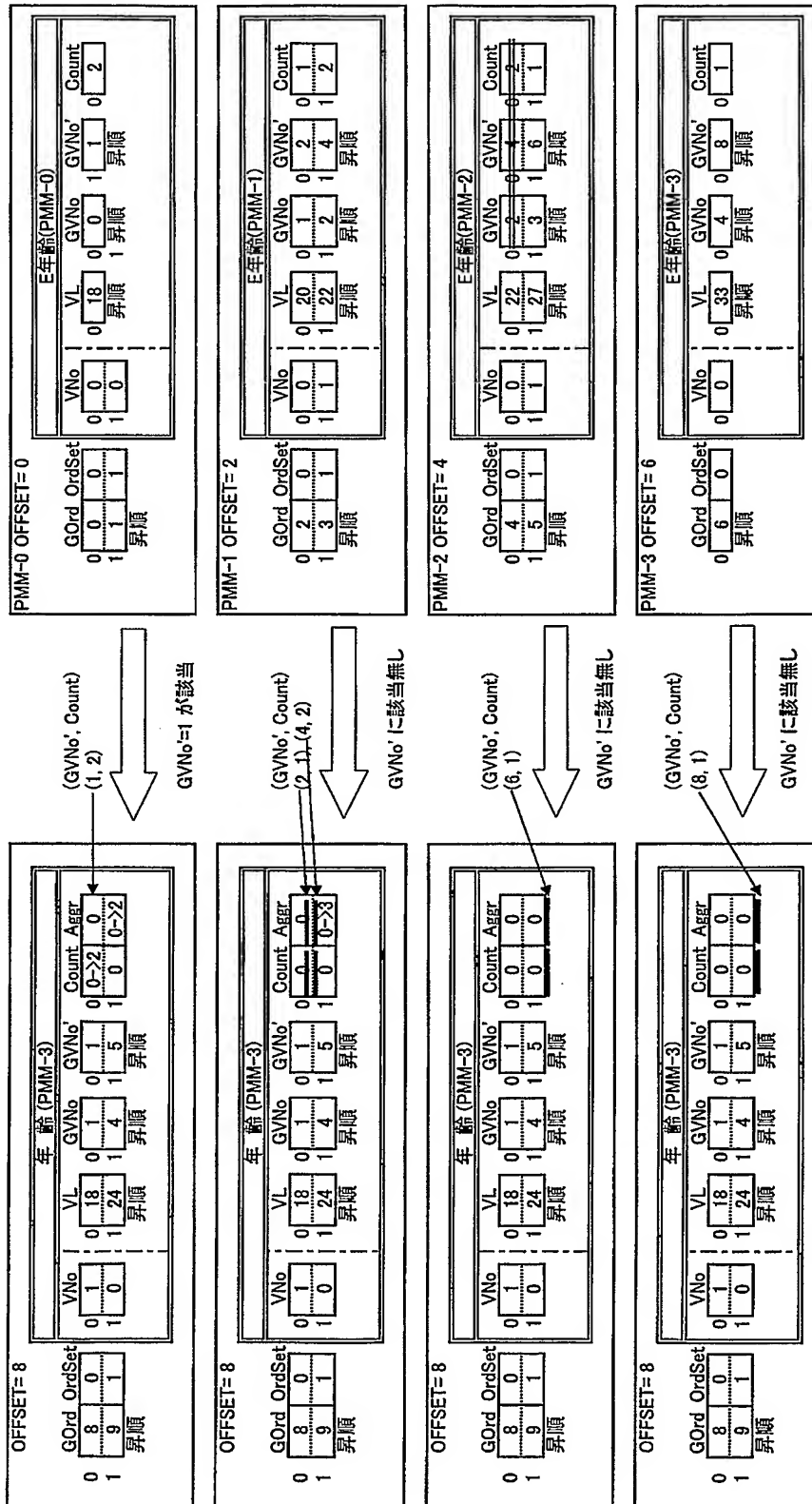
【図 37】

図37

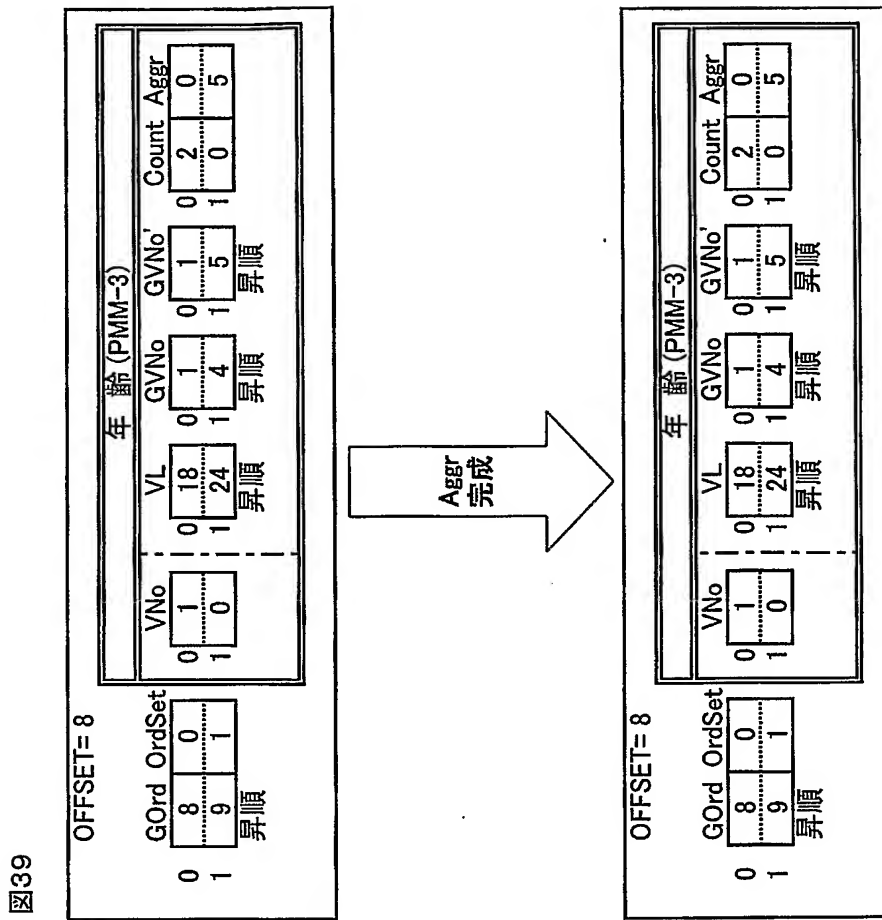


【図 38】

図38



【図 39】



【図 40】

図40

OFFSET=0

SetAggr GOrd OrdSet									
0	0	0	0	0	0	0	159	0	0
1	2	1	1	1	1	1	168	1	2
2	2	2	2	2	2	2	172	2	64
昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順	
性別 (PMM-0)		年齢 (PMM-0)		身長 (PMM-0)		体重 (PMM-0)			
VNo	0 1	VL	0 男	GVNo	0 0	VNo	0 1	VL	0 48
	1 0		1 女		1 1		1 2		1 55
	2 1		2 昇順		2 2		2 0		2 64
昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順	

OFFSET=3

SetAggr GOrd OrdSet									
0	2	3	0	0	0	0	172	0	4
1	2	4	1	1	5	1	181	1	7
2	2	4	1	2	7	2		2	
昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順	
性別 (PMM-1)		年齢 (PMM-1)		身長 (PMM-1)		体重 (PMM-1)			
VNo	0 1	VL	0 男	GVNo	0 0	VNo	0 0	VL	0 48
	1 0		1 女		1 1		1 1		1 78
			2 昇順		2 2				2 昇順
昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順	

OFFSET=5

SetAggr GOrd OrdSet									
0	2	5	0	0	0	0	166	0	1
1	3	6	1	1	2	1	168	1	2
2	3	7	2	2	8	2	174	2	5
昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順	
性別 (PMM-2)		年齢 (PMM-2)		身長 (PMM-2)		体重 (PMM-2)			
VNo	0 1	VL	0 男	GVNo	0 0	VNo	0 1	VL	0 52
	1 1		1 女		1 1		1 0		1 55
	2 0		2 昇順		2 2		2 2		2 65
昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順	

OFFSET=8

SetAggr GOrd OrdSet									
0	4	8	0	0	0	0	170	0	3
1	4	9	1	1	4	1	177	1	6
2	4	9	1	2	5	2		2	
昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順	
性別 (PMM-3)		年齢 (PMM-3)		身長 (PMM-3)		体重 (PMM-3)			
VNo	0 0	VL	0 男	GVNo	0 0	VNo	0 1	VL	0 55
	1 1		1 女		1 1		1 0		1 64
			2 昇順		2 2				2 昇順
昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順		昇順 昇順	

【図 4 1】

図41

PMM-0				
	性別	年齢	身長	体重
0	女	18	168	55
1	男	21	172	64
2	女	24	159	48

PMM-1				
	性別	年齢	身長	体重
0	女	16	172	48
1	男	28	181	78

PMM-2				
	性別	年齢	身長	体重
0	女	20	166	55
1	女	16	168	52
2	男	33	174	65

PMM-3				
	性別	年齢	身長	体重
0	男	24	177	64
1	女	18	170	55

【図 42】

図42

OFFSET= 0

GOrd	OrdSet
0	0
1	1

昇順

E年齢(PMM-0)				
VNo	VL	GVNo	GVNo'	Count
0	0	0	0	1
1	0	1	1	2

昇順

イベント(PMM-0)		
VNo	VL	GVNo
0	1	1
1	0	3

昇順

OFFSET= 2

GOrd	OrdSet
0	2
1	3

昇順

E年齢(PMM-1)				
VNo	VL	GVNo	GVNo'	Count
0	0	1	2	1
1	1	2	4	1

昇順

イベント(PMM-1)		
VNo	VL	GVNo
0	0	0
1	1	2

昇順

OFFSET= 4

GOrd	OrdSet
0	4
1	5

昇順

E年齢(PMM-2)				
VNo	VL	GVNo	GVNo'	Count
0	0	2	4	1
1	1	3	6	1

昇順

イベント(PMM-2)		
VNo	VL	GVNo
0	0	5
1	1	6

昇順

OFFSET= 6

GOrd	OrdSet
0	6
1	0

昇順

E年齢(PMM-3)				
VNo	VL	GVNo	GVNo'	Count
0	0	4	8	1

昇順

イベント(PMM-3)		
VNo	VL	GVNo
0	0	4

昇順

【図 43】

図43

PMM-0		OFFSET = 0	
0 1	E年齢	イベント	
	18	H2	
	18	C1	

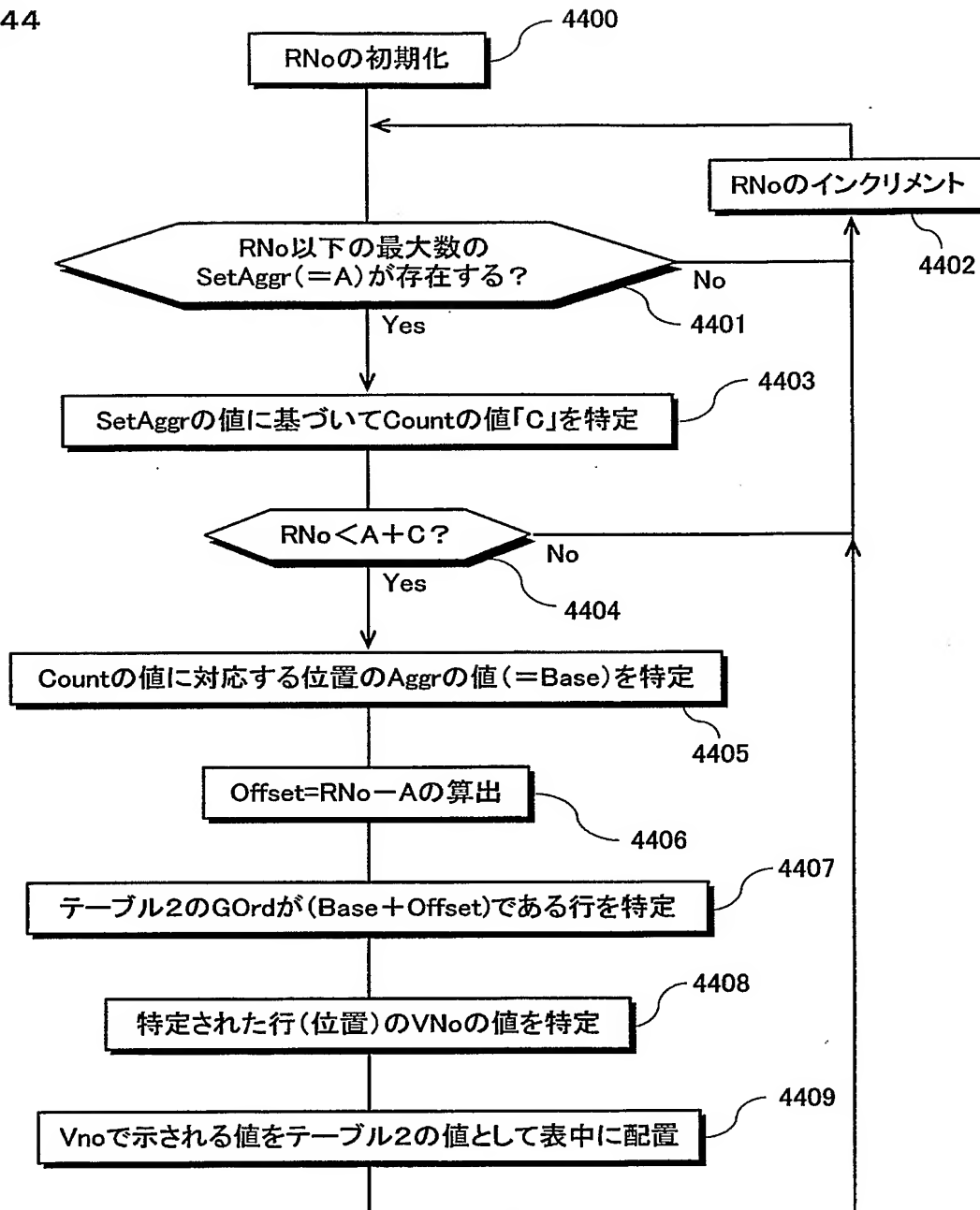
PMM-1		OFFSET = 2	
0 1	E年齢	イベント	
	20	A	
	22	C2	

PMM-2		OFFSET = 4	
0 1	E年齢	イベント	
	22	P1	
	27	P2	

PMM-3		OFFSET = 6	
0 1	E年齢	イベント	
	33	M	

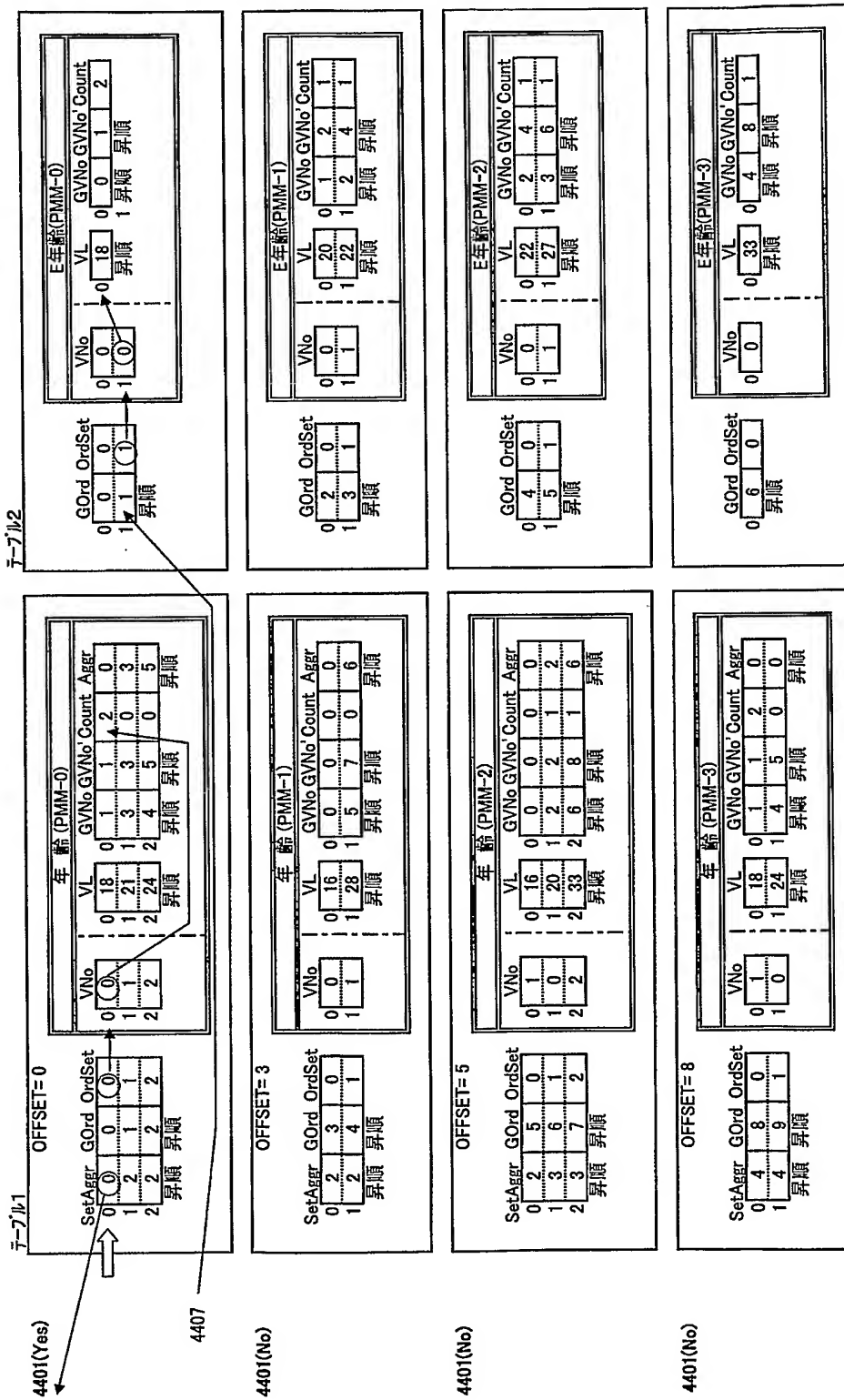
【図 4 4】

図 44

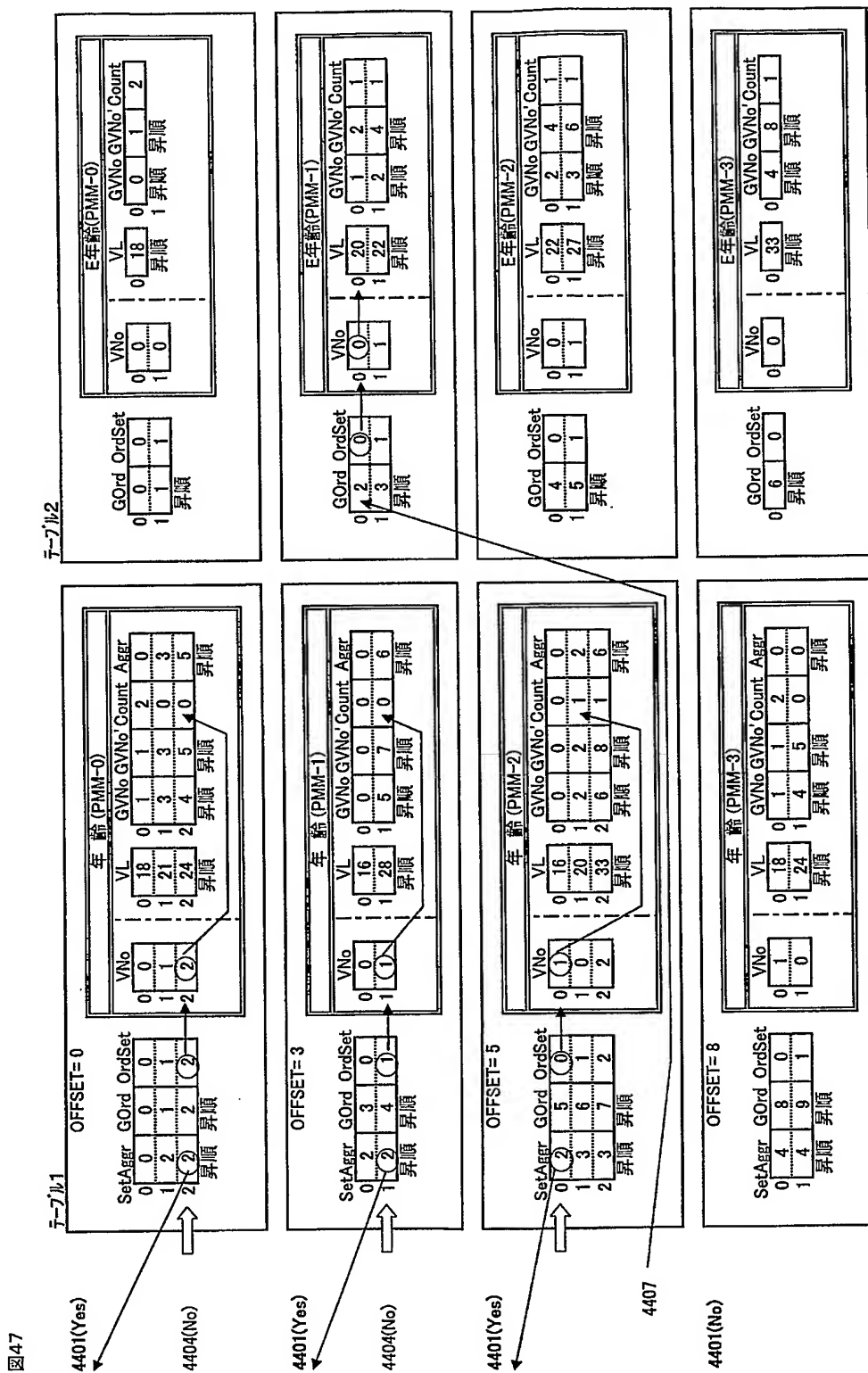


【図 46】

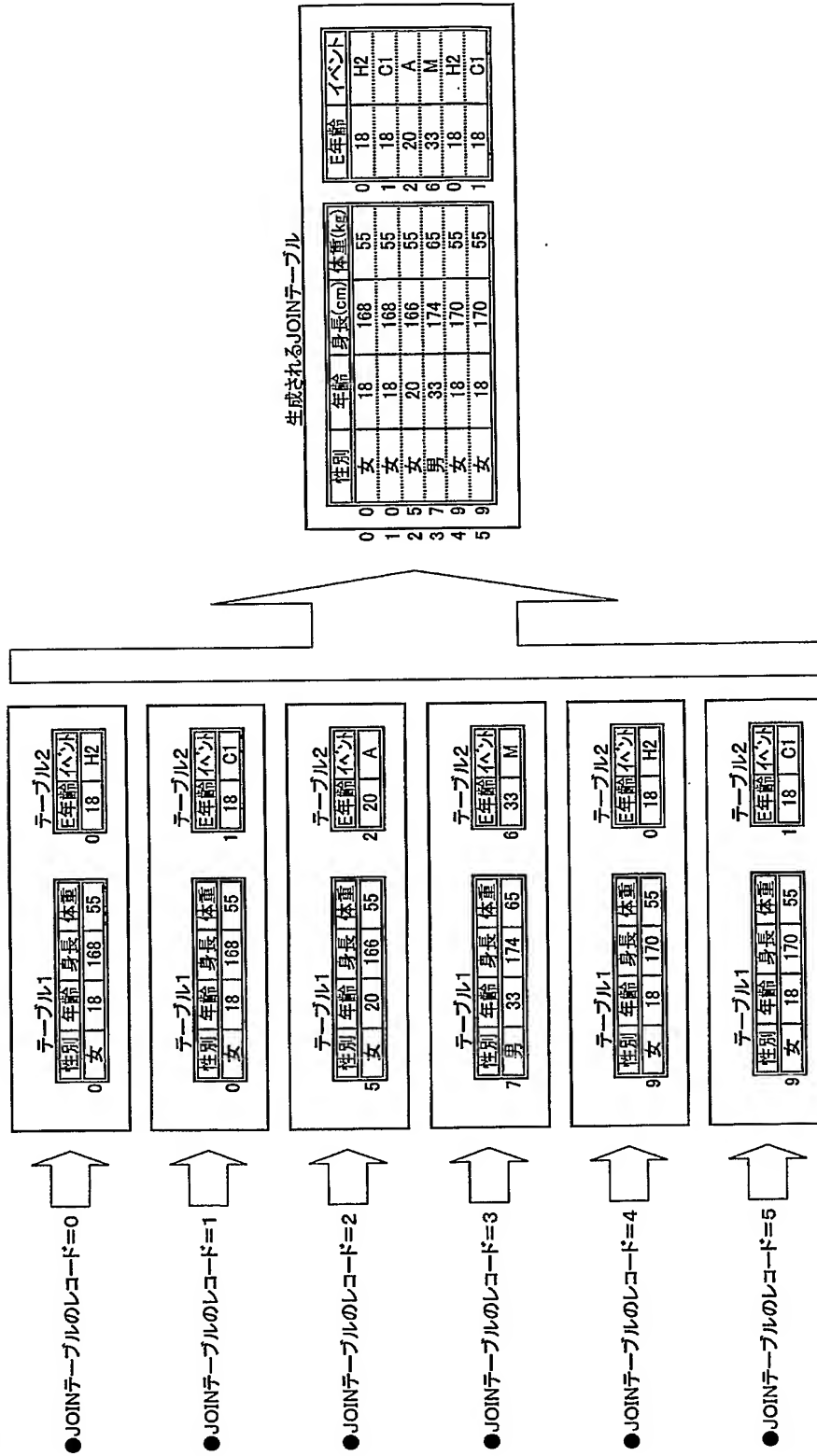
図46



【図 47】



【図 52】



【図 53】

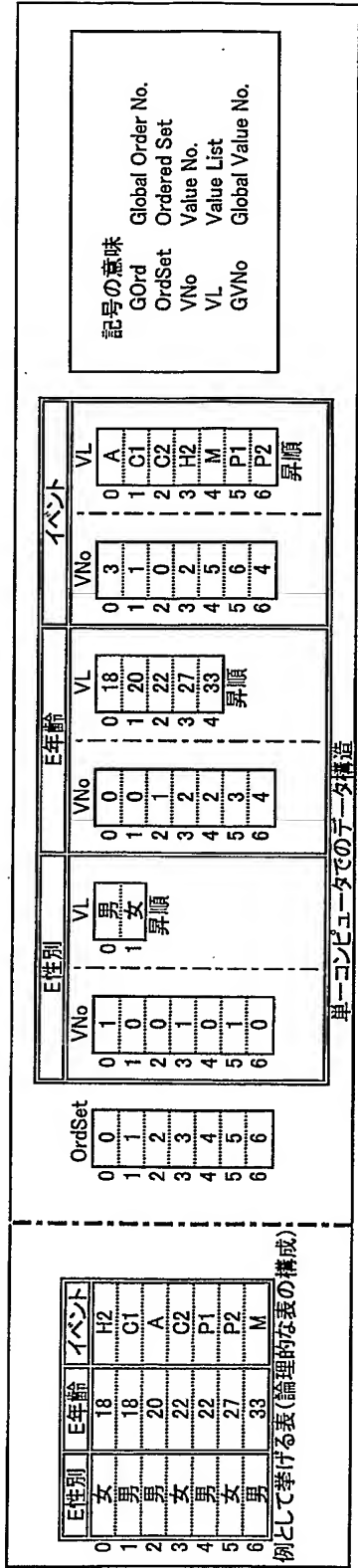


図53

【図 54】

図54

PMM-0

OFFSET = 0

	E性別	E年齢	イベント
0	女	18	H2
1	男	18	C1

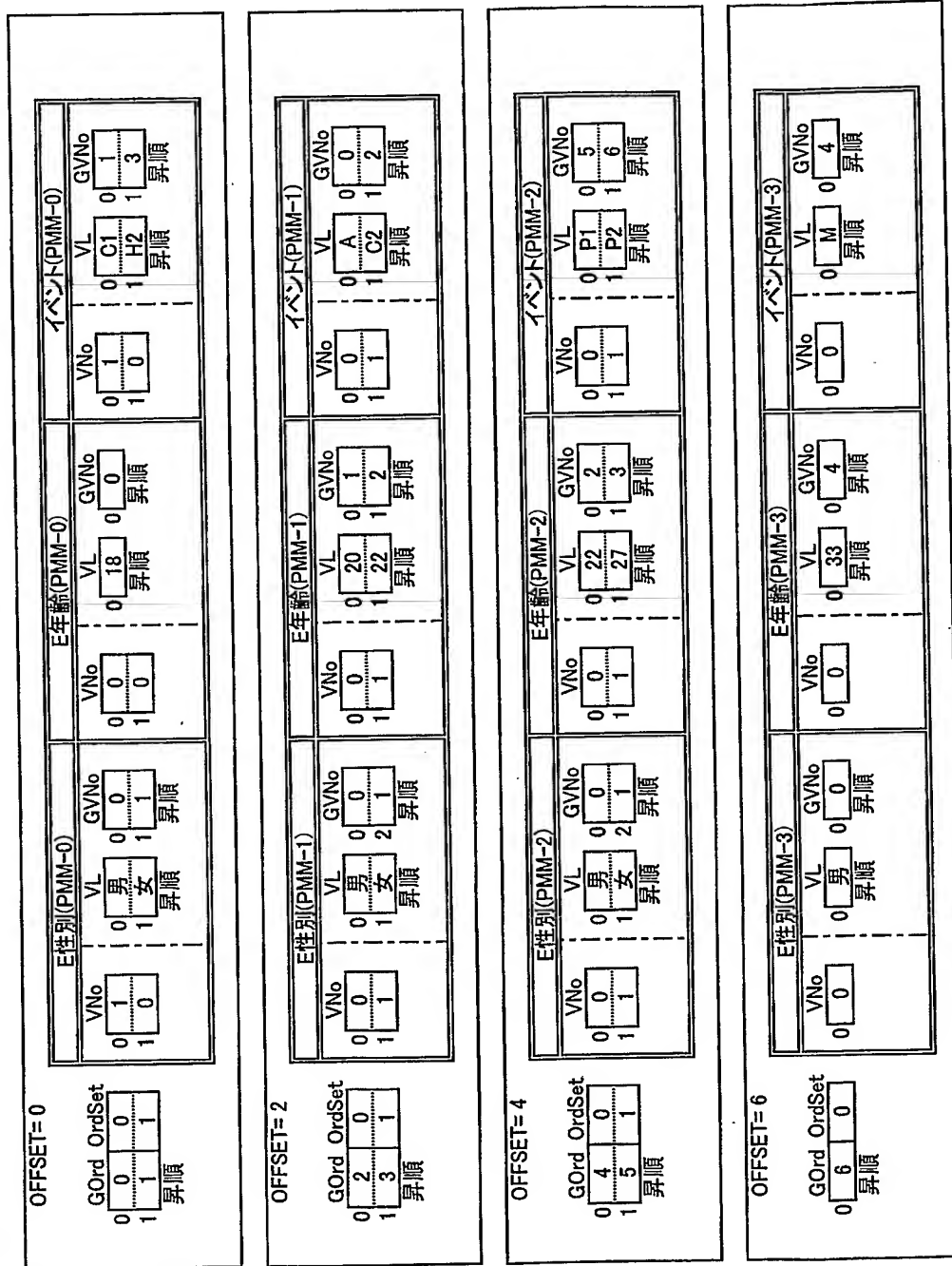
PMM-1		OFFSET = 2	
	E性別	E年齢	イベント
0	男	20	A
1	女	22	C2

PMM-2		OFFSET = 4	
	E性別	E年齢	イベント
0	男	22	P1
1	女	27	P2

PMM-3		OFFSET = 6	
	E性別	E年齢	イベント
0	男	33	M

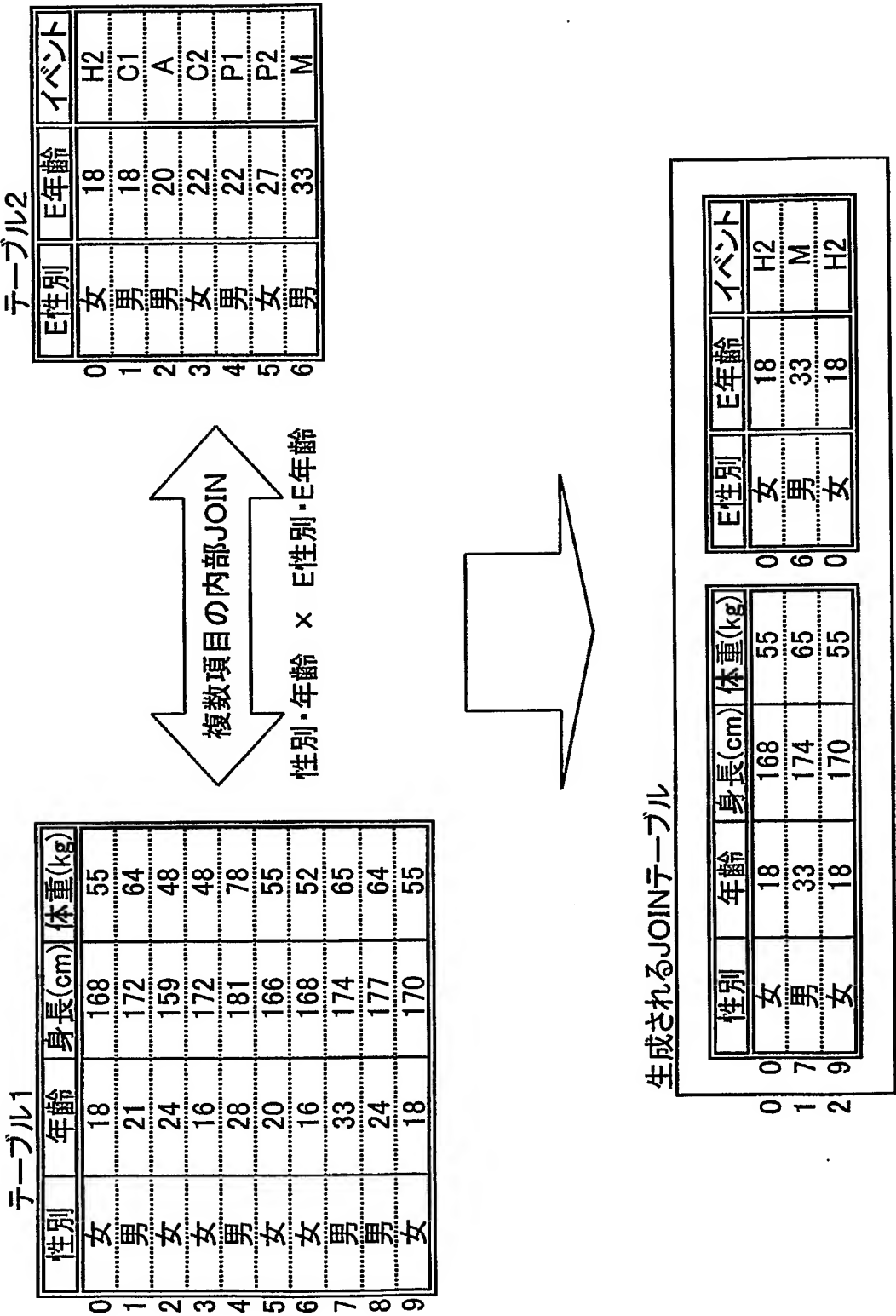
【図 55】

図55



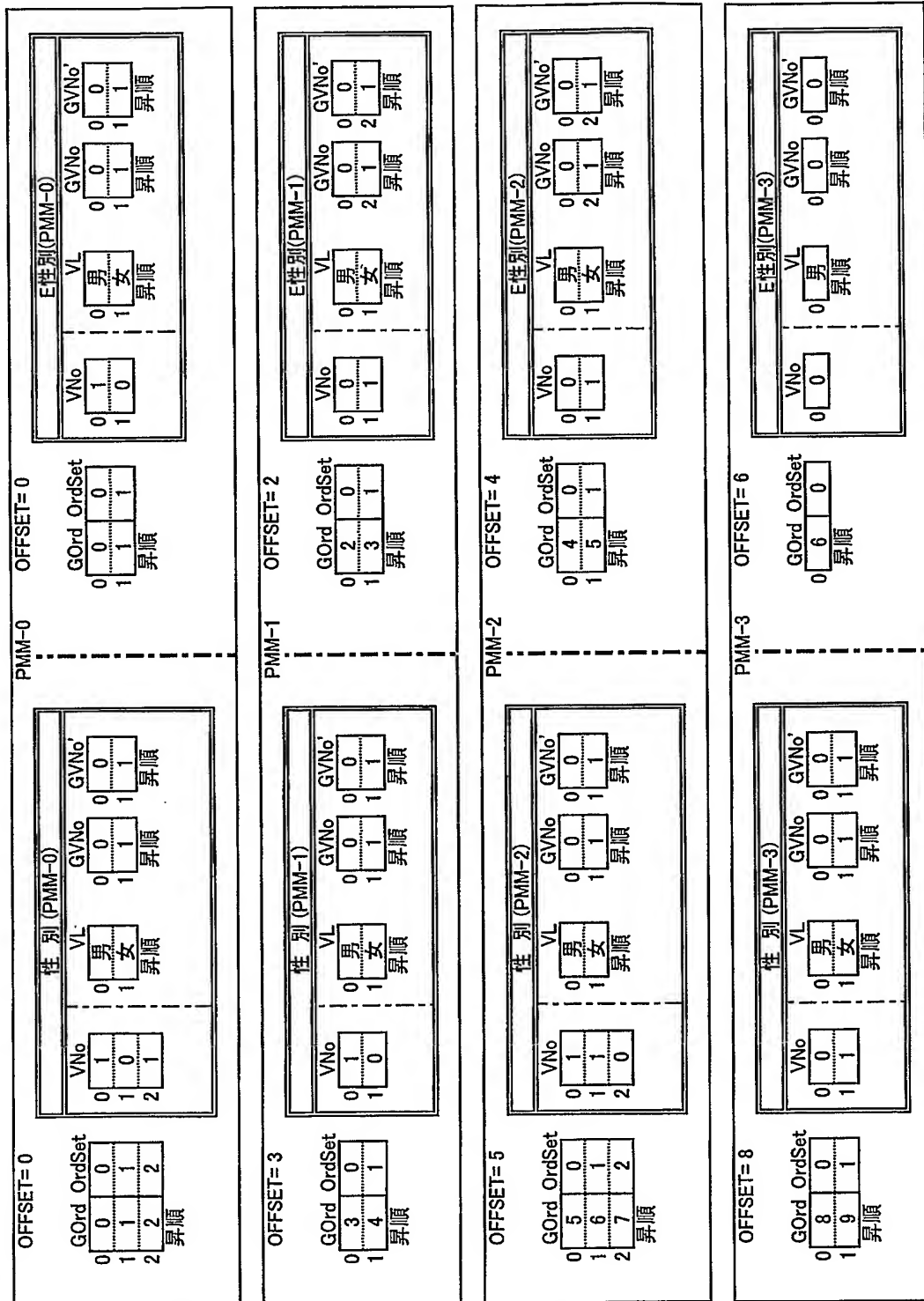
【図 56】

図56



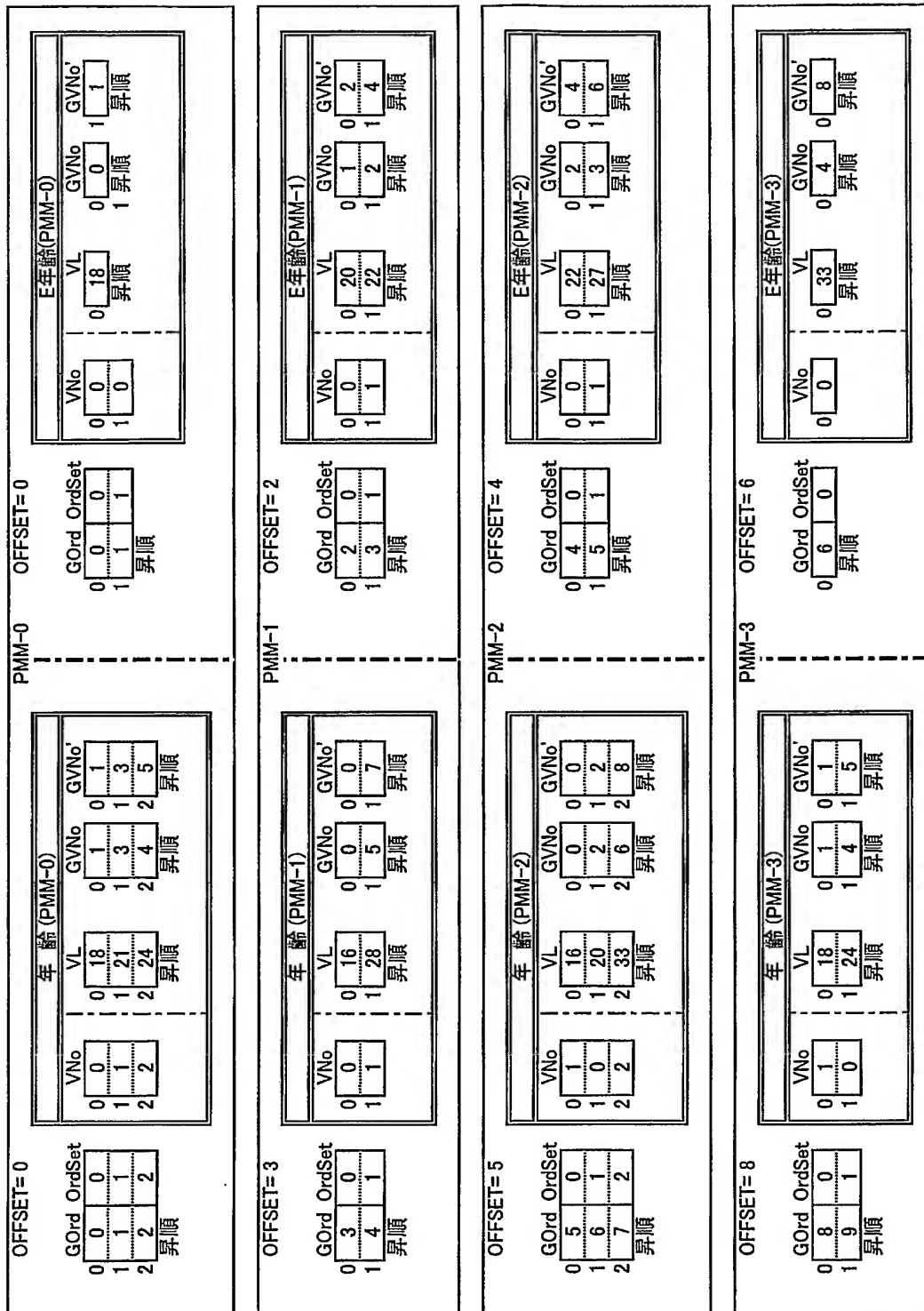
【図 57】

図57



【図 58】

図58



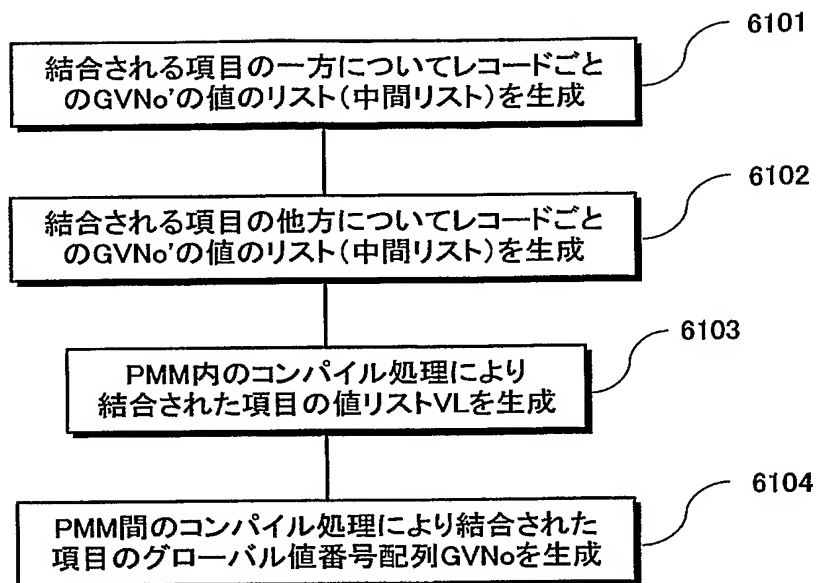
【図 60】

図60

PMM-0 OFFSET= 0									
GOrd OrdSet		E性別(PMM-0)				E年齢(PMM-0)			
0	0	VNo	VL	GVNo	GVNo'	VNo	VL	GVNo	GVNo'
1	1	1	男	0	0	0	18	0	1
		0	女	1	1	1	昇順	1	昇順
		1	昇順		昇順				昇順
PMM-1 OFFSET= 2									
GOrd OrdSet		E性別(PMM-1)				E年齢(PMM-1)			
0	2	VNo	VL	GVNo	GVNo'	VNo	VL	GVNo	GVNo'
1	3	0	男	0	0	0	20	1	2
		1	女	1	1	1	22	2	4
			昇順		昇順		昇順		昇順
PMM-2 OFFSET= 4									
GOrd OrdSet		E性別(PMM-2)				E年齢(PMM-2)			
0	4	VNo	VL	GVNo	GVNo'	VNo	VL	GVNo	GVNo'
1	5	0	男	0	0	0	22	2	4
		1	女	1	1	1	27	3	6
			昇順		昇順		昇順		昇順
PMM-3 OFFSET= 6									
GOrd OrdSet		E性別(PMM-3)				E年齢(PMM-3)			
0	6	VNo	VL	GVNo	GVNo'	VNo	VL	GVNo	GVNo'
		0	男	0	0	0	33	4	8
			昇順		昇順		昇順		昇順

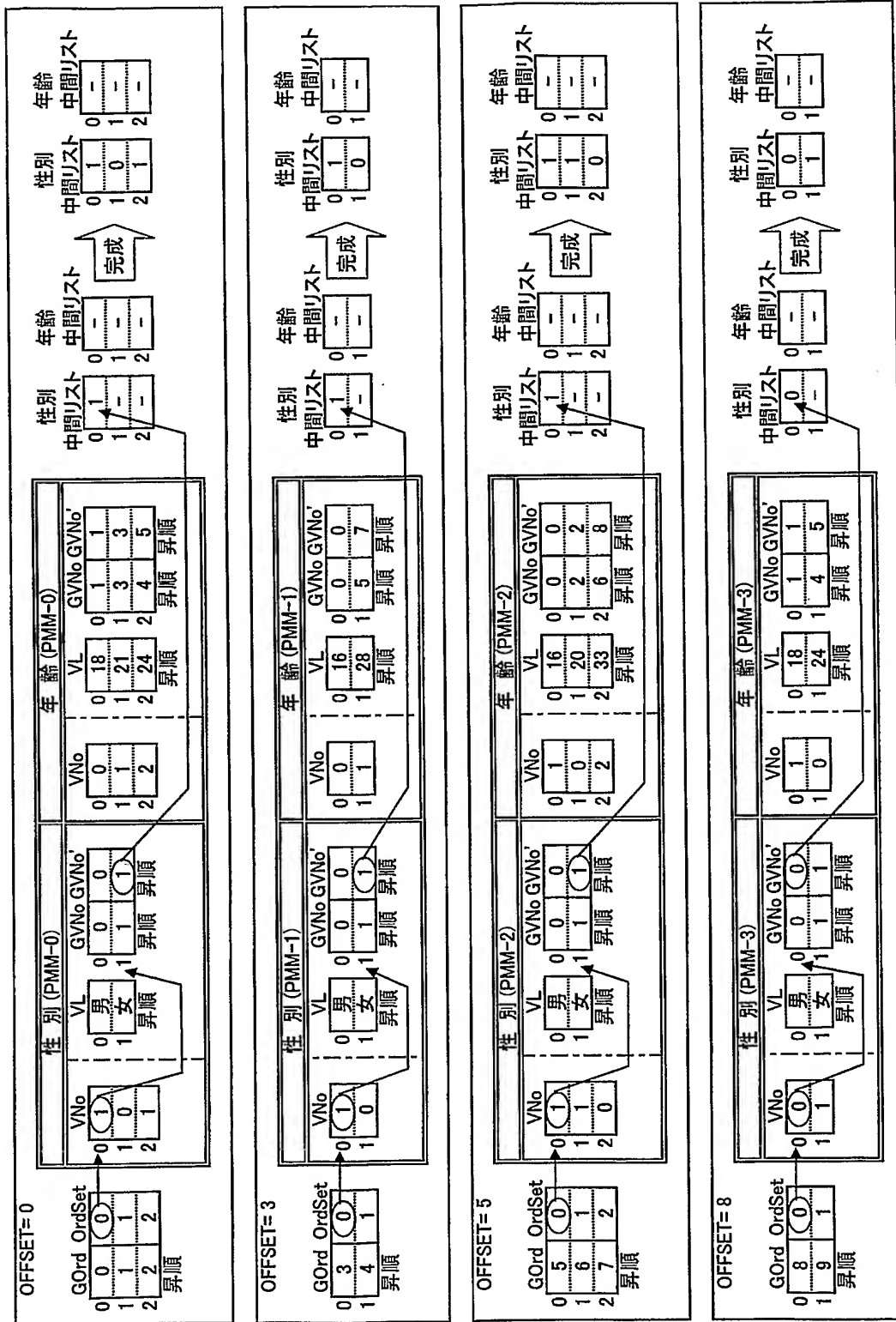
【図 61】

図61



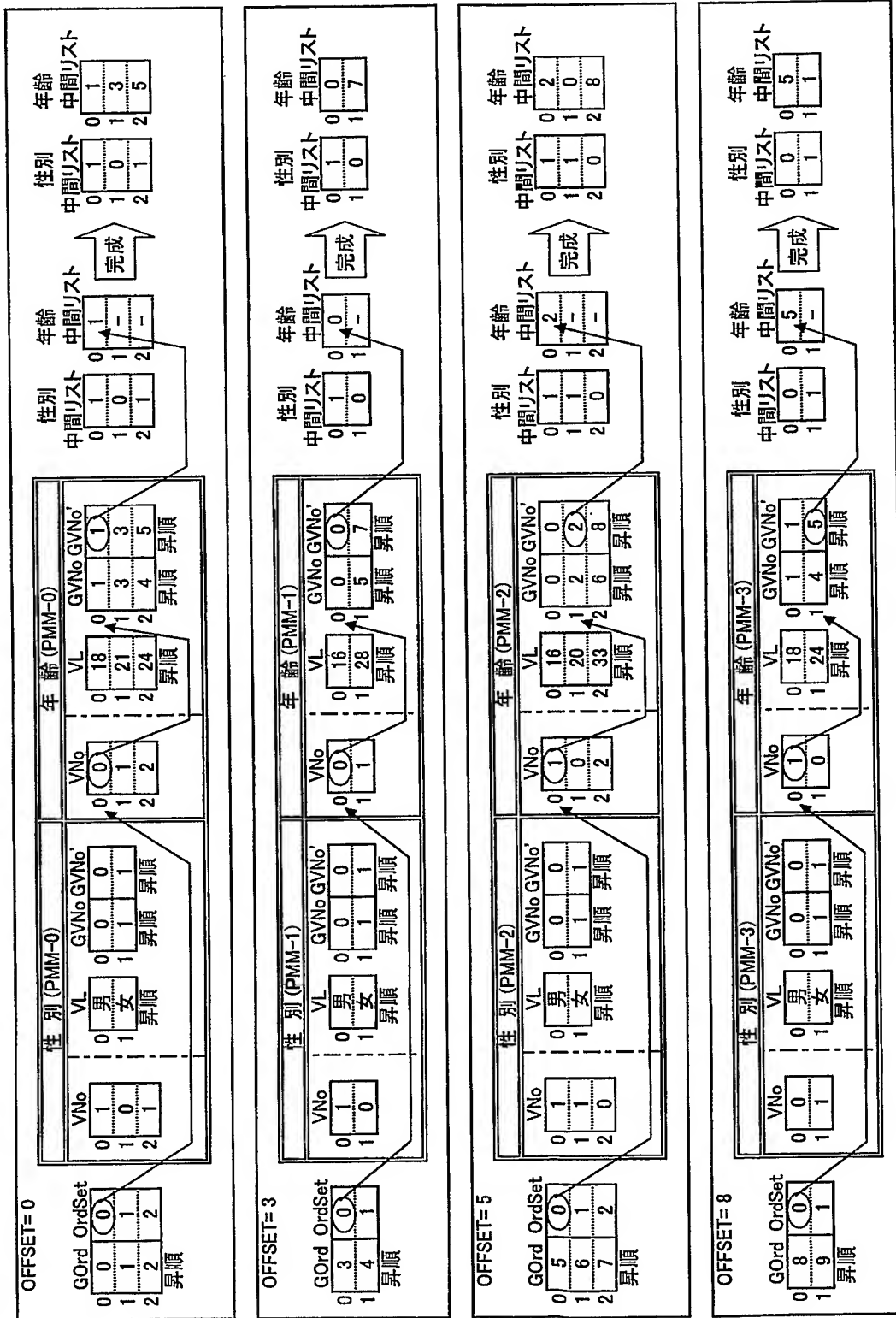
【図 62】

図62



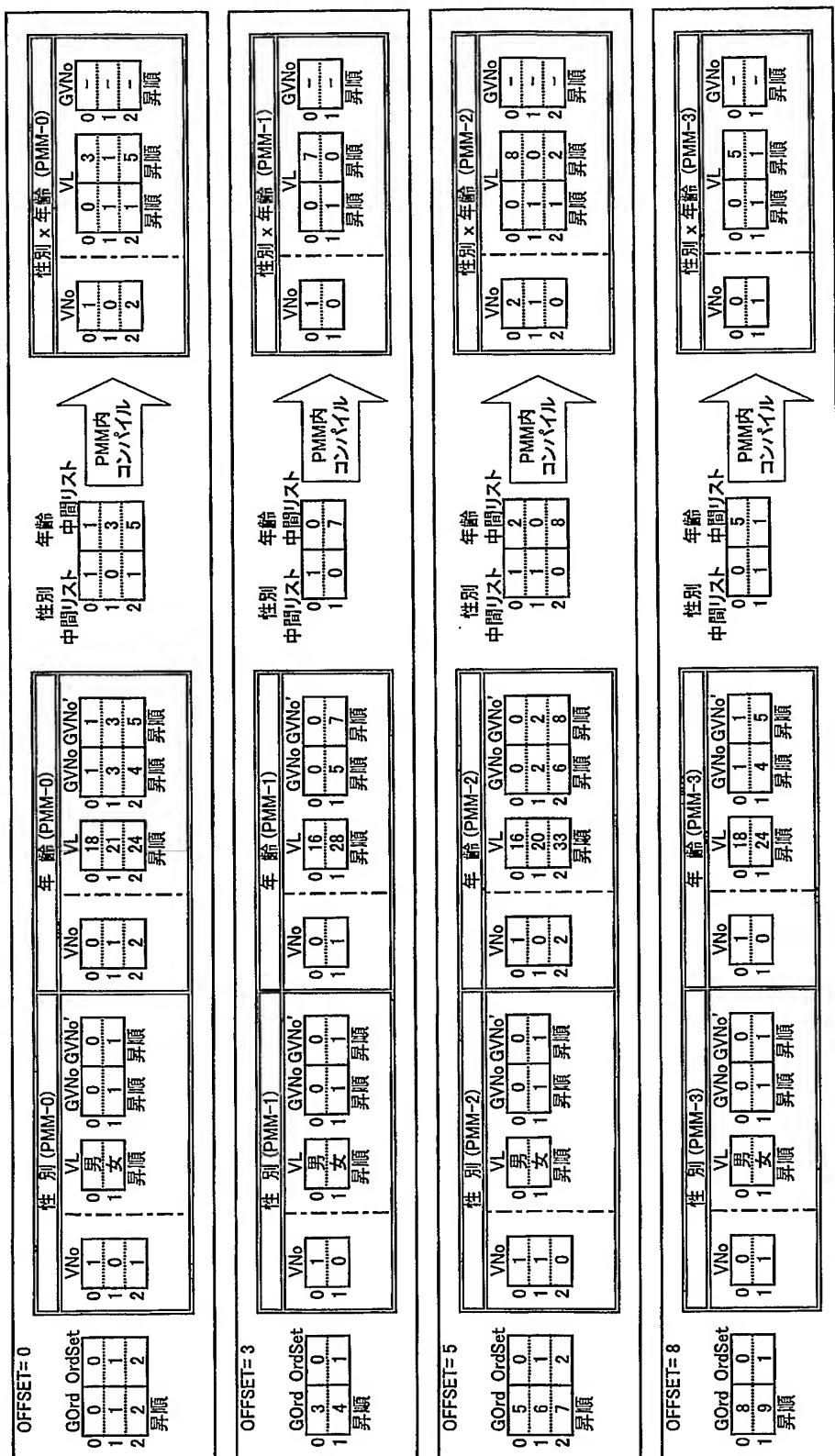
【図 63】

図63



【図 64】

図64



【図 65】

図65

性別 × 年齢 (PMM-0)					
VNo		VL		GVNo	
0	1	0	0 3	0	0
1	0	1	1 1	1	5
2	2	2	1 5	2	7
			昇順 昇順		昇順

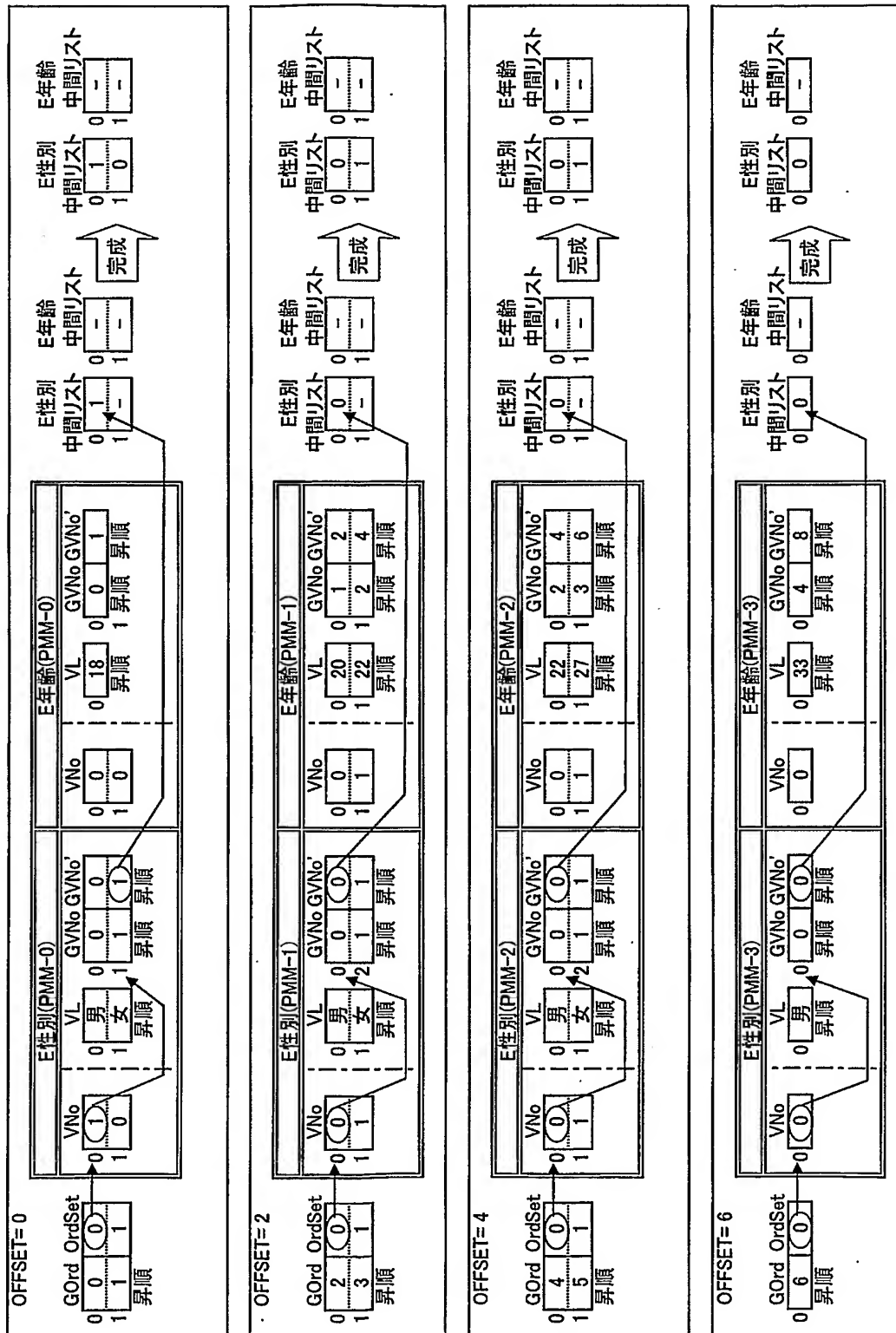
性別 × 年齢 (PMM-1)					
VNo		VL		GVNo	
0	1	0	0 7	0	2
1	0	1	1 0	1	4
			昇順 昇順		昇順

性別 × 年齢 (PMM-2)					
VNo		VL		GVNo	
0	2	0	0 8	0	3
1	1	1	1 0	1	4
2	0	2	1 2	2	6
			昇順 昇順		昇順

性別 × 年齢 (PMM-3)					
VNo		VL		GVNo	
0	0	0	0 5	0	1
1	1	1	1 1	1	5
			昇順 昇順		昇順

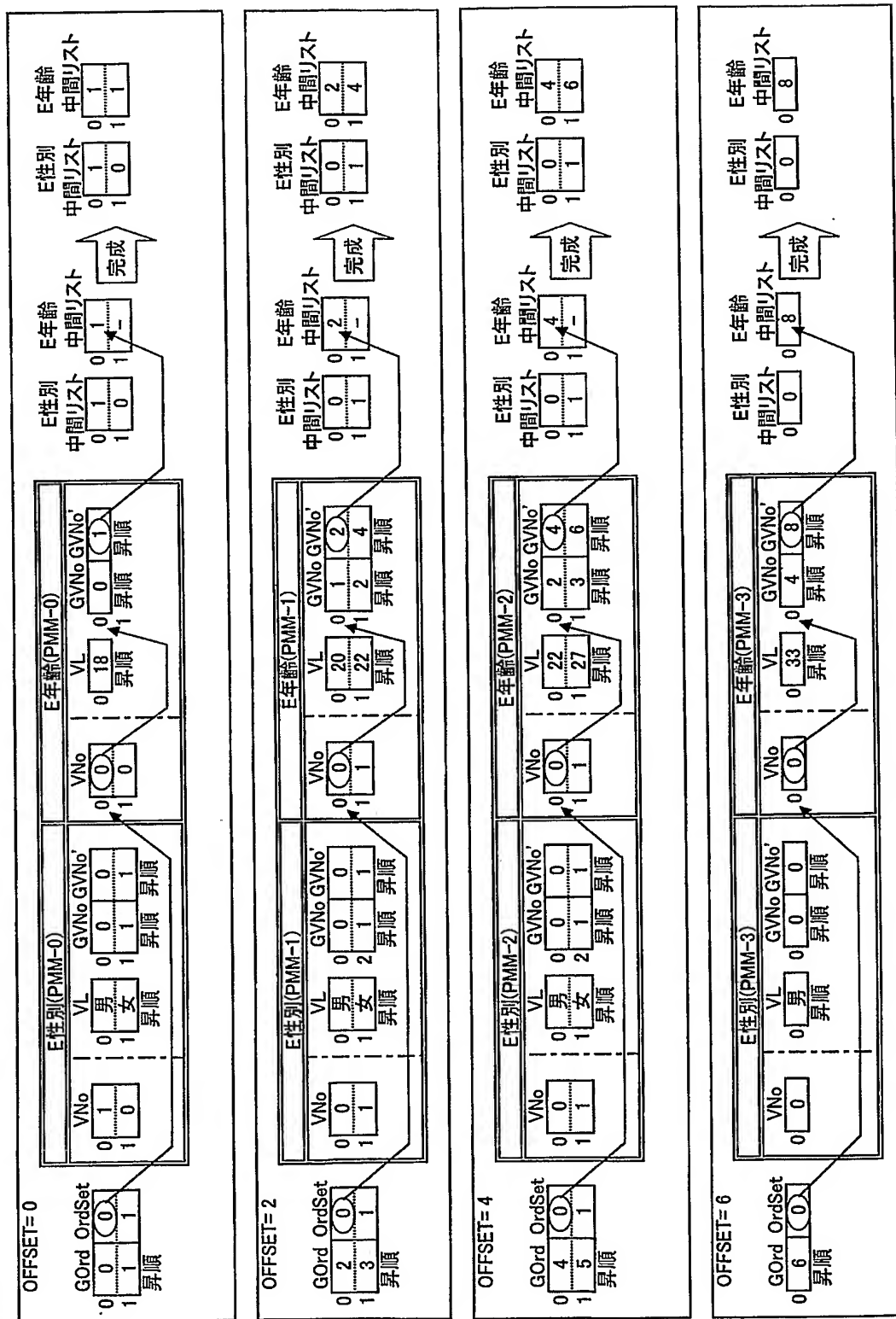
【図 66】

図66



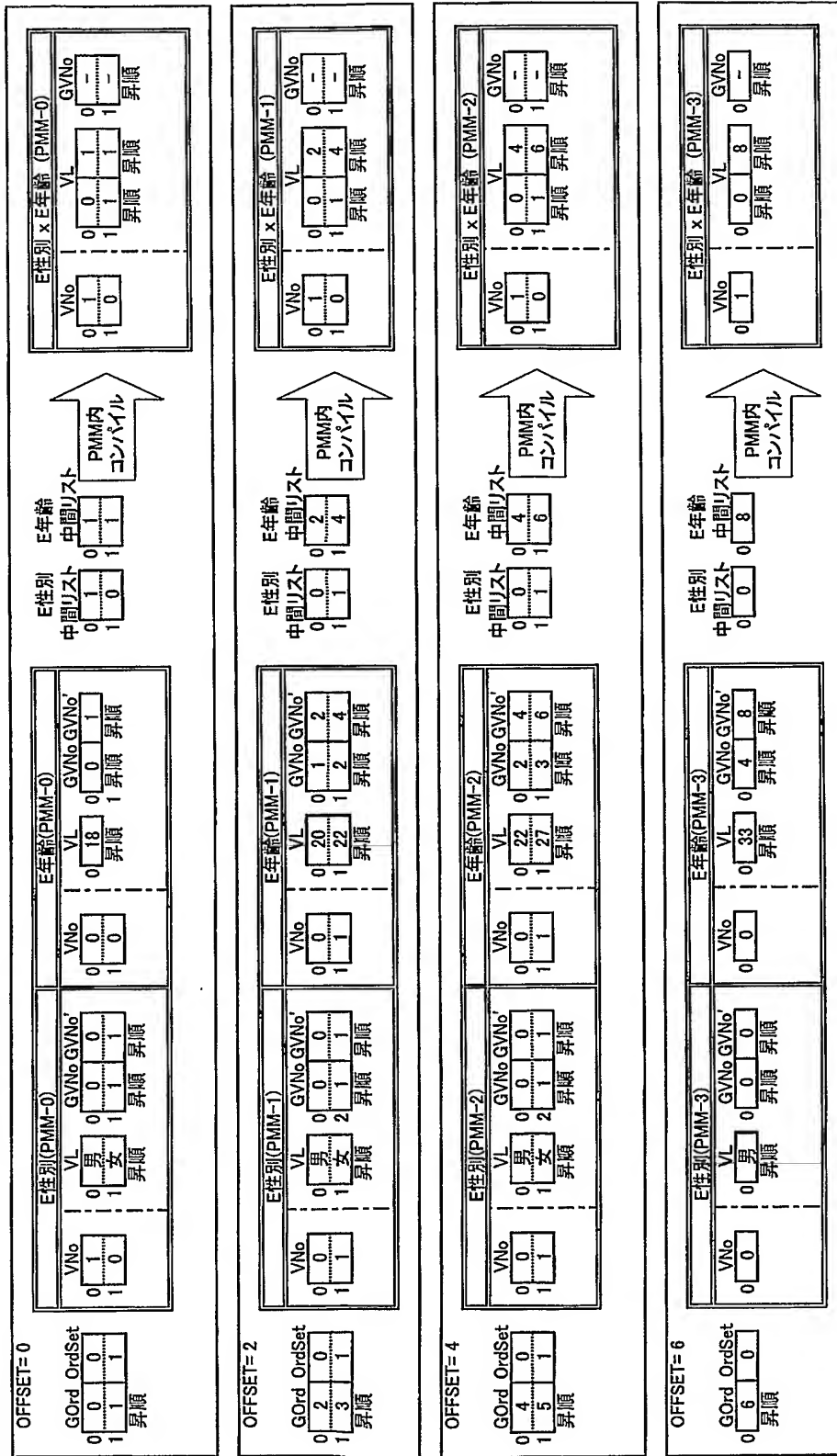
【図 67】

図67



【図 68】

図 68



【図 69】

図69

E性別 x E年齢 (PMM-0)									
VNo		VL				GVNo			
0	1	0	0	1		0	0		
1	0	1	1	1		1	4		
		昇順		昇順		昇順			

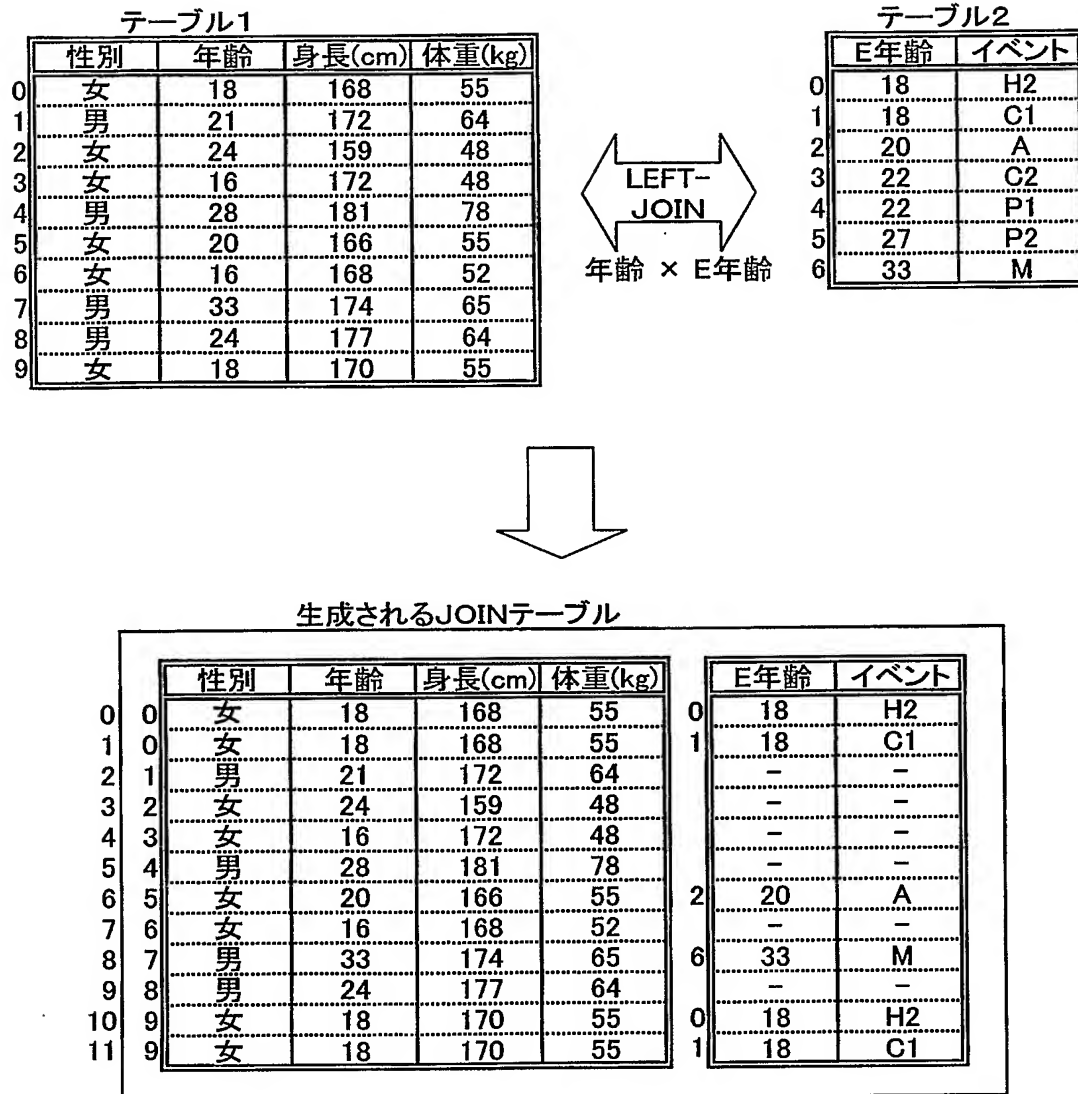
E性別 x E年齢 (PMM-1)									
VNo		VL				GVNo			
0	1	0	0	2		0	1		
1	0	1	1	4		1	5		
		昇順		昇順		昇順			

E性別 x E年齢 (PMM-2)									
VNo		VL				GVNo			
0	1	0	0	4		0	2		
1	0	1	1	6		1	6		
		昇順		昇順		昇順			

E性別 x E年齢 (PMM-3)									
VNo		VL				GVNo			
0	1	0	0	8		0	3		
		昇順		昇順		昇順			

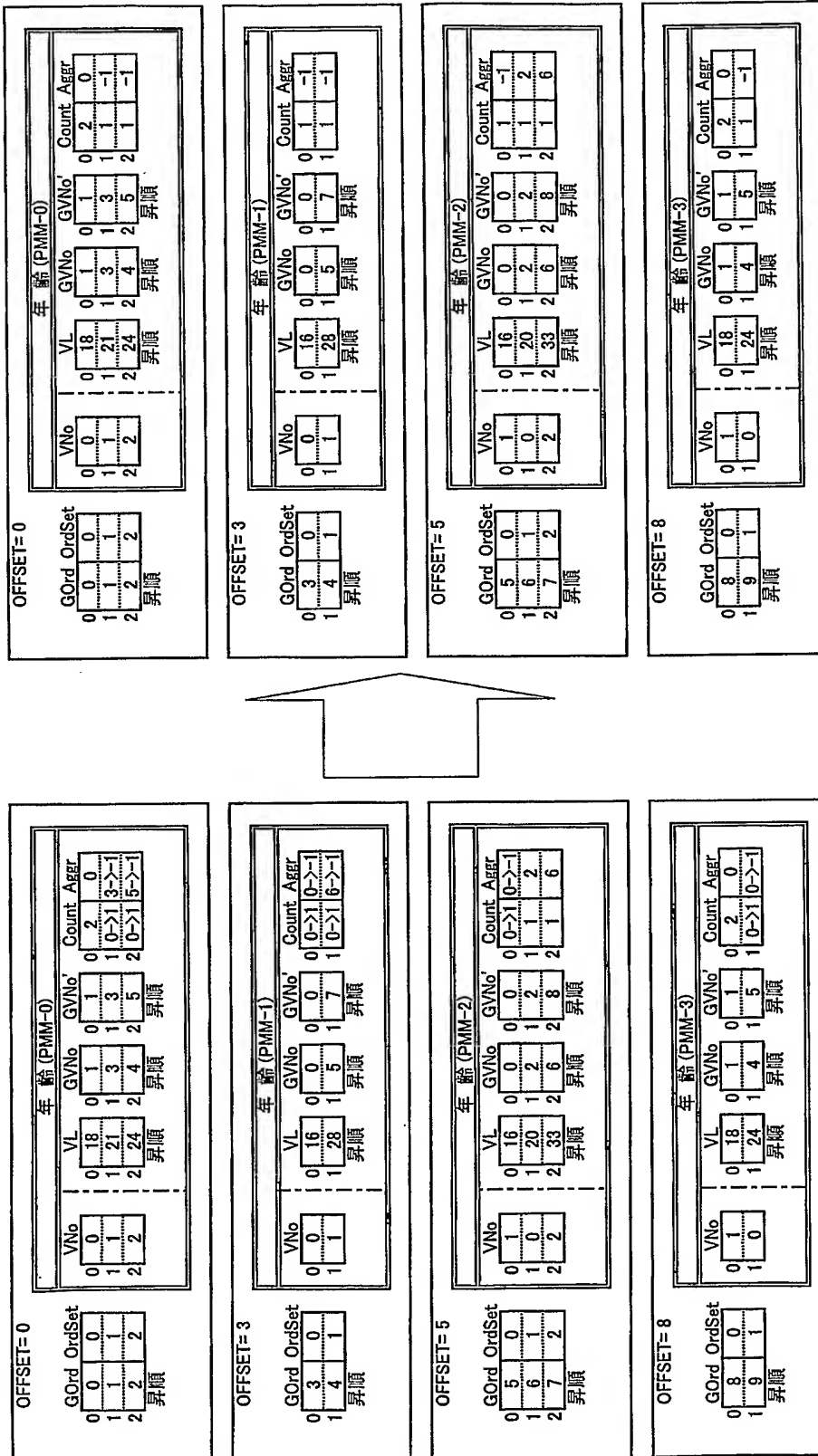
【図 70】

図70



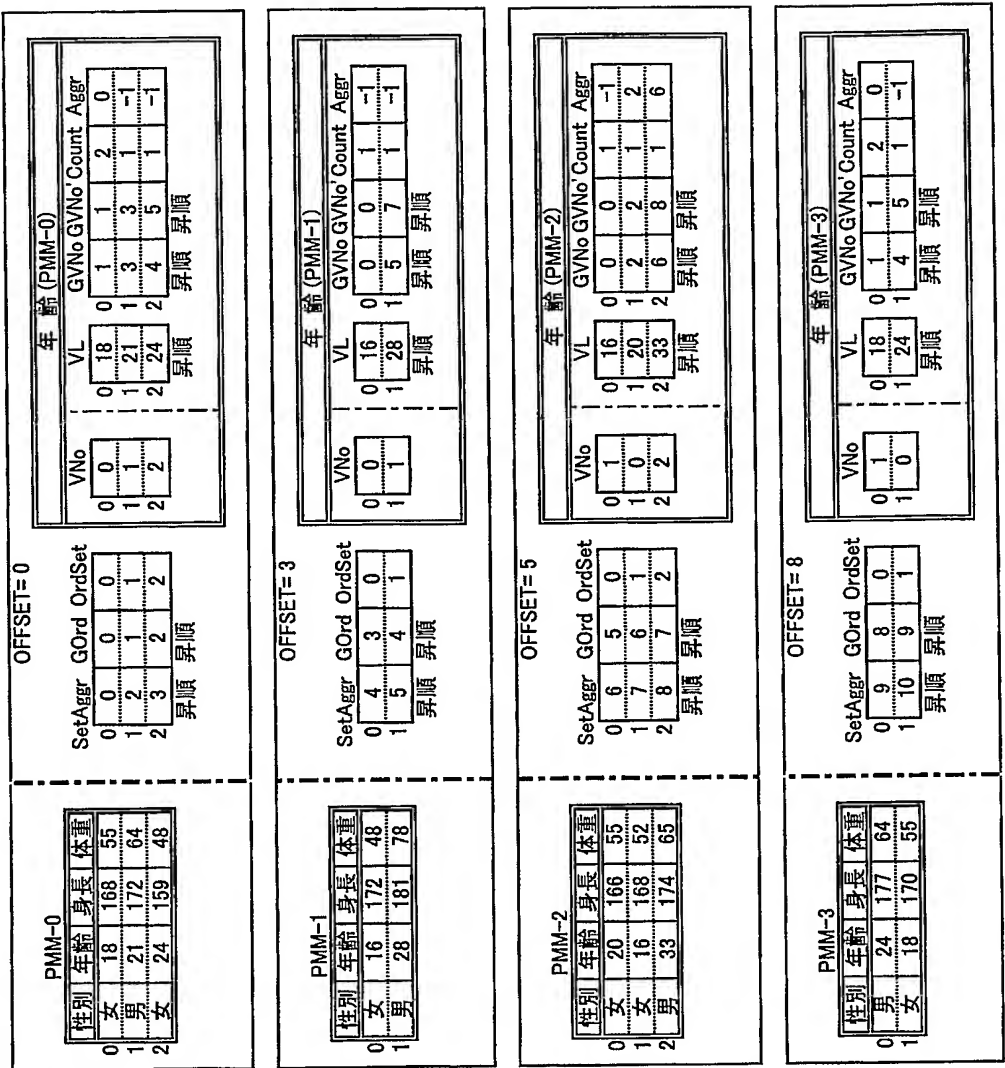
【図 71】

図71



【図 72】

図 72



0 0	1 1	2 2
-----	-----	-----

VL

0 16	1 28
------	------

昇順

SetAggr

GOrd

OrdSet

0 4	3 0	0 1
1 5	4 4	1 1

昇順

OFFSET=3

0 4	3 0	0 1
1 5	4 4	1 1

0 0	1 1	2 2
-----	-----	-----

VNo

0 0	1 1
-----	-----

0 0	1 1	2 2
-----	-----	-----

VL

0 16	1 20	2 33
------	------	------

昇順

SetAggr

GOrd

OrdSet

0 6	5 0	0 0
1 7	6 1	1 1
2 8	7 2	2 2

昇順

OFFSET=5

0 6	5 0	0 0
1 7	6 1	1 1
2 8	7 2	2 2

0 0	1 1	2 2
-----	-----	-----

VNo

0 1	0 0	2 2
-----	-----	-----

0 0	1 1	2 2
-----	-----	-----

VL

0 18	1 24
------	------

昇順

SetAggr

GOrd

OrdSet

0 9	8 0	0 0
1 10	9 1	1 1

昇順

OFFSET=8

0 9	8 0	0 0
1 10	9 1	1 1

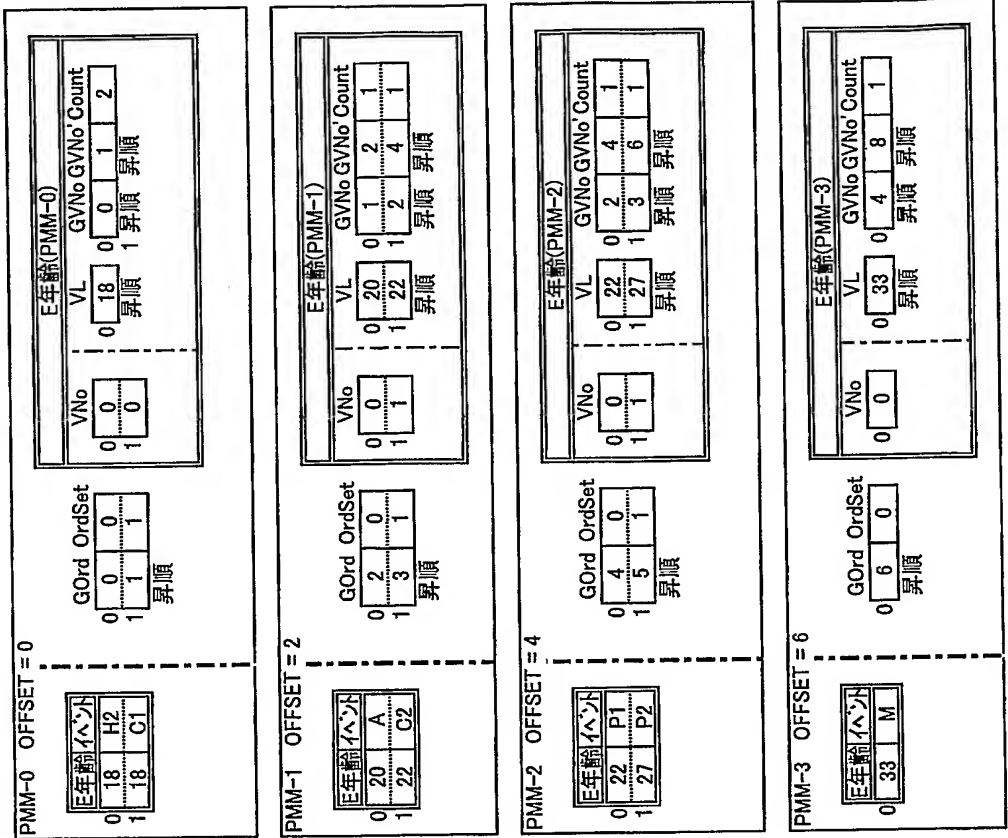
0 0	1 1	2 2
-----	-----	-----

VNo

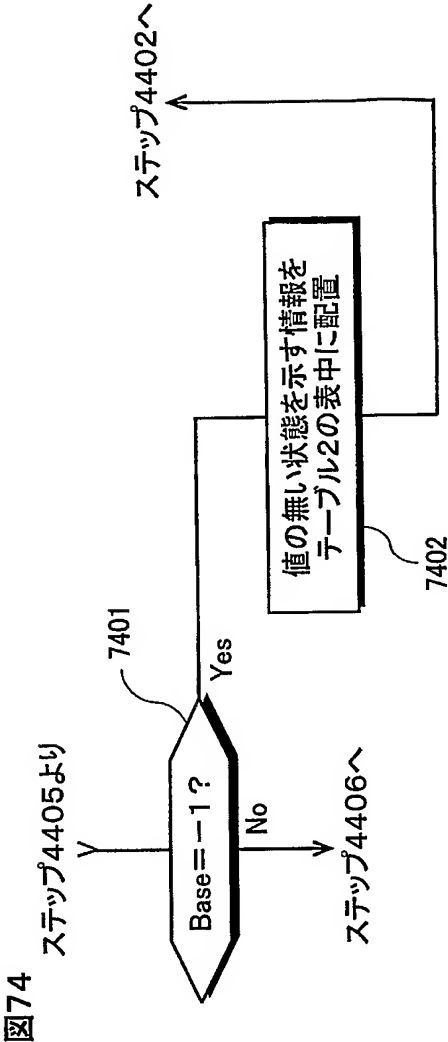
0 1	0 0
-----	-----

【図 73】

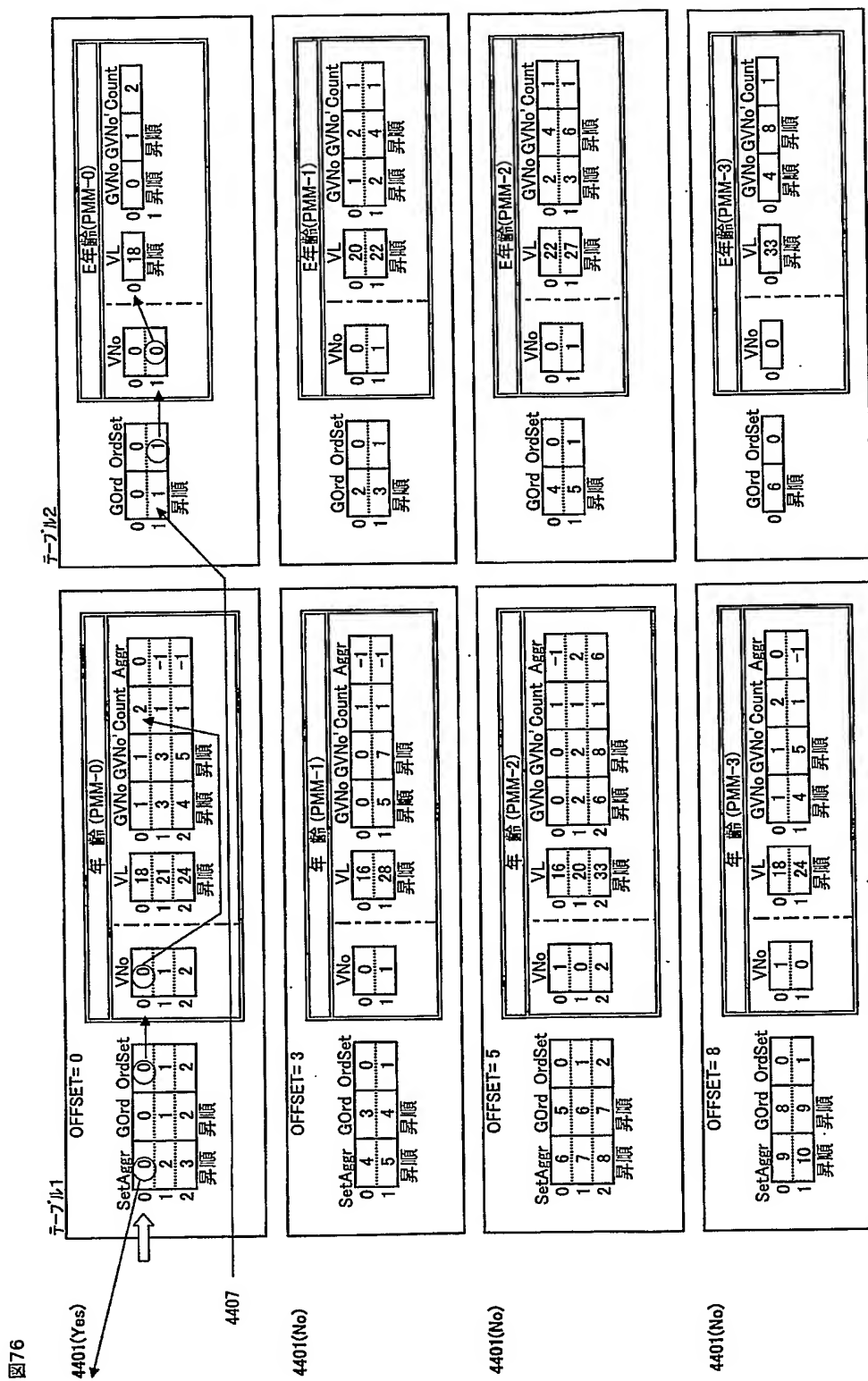
図73



【図 7 4】

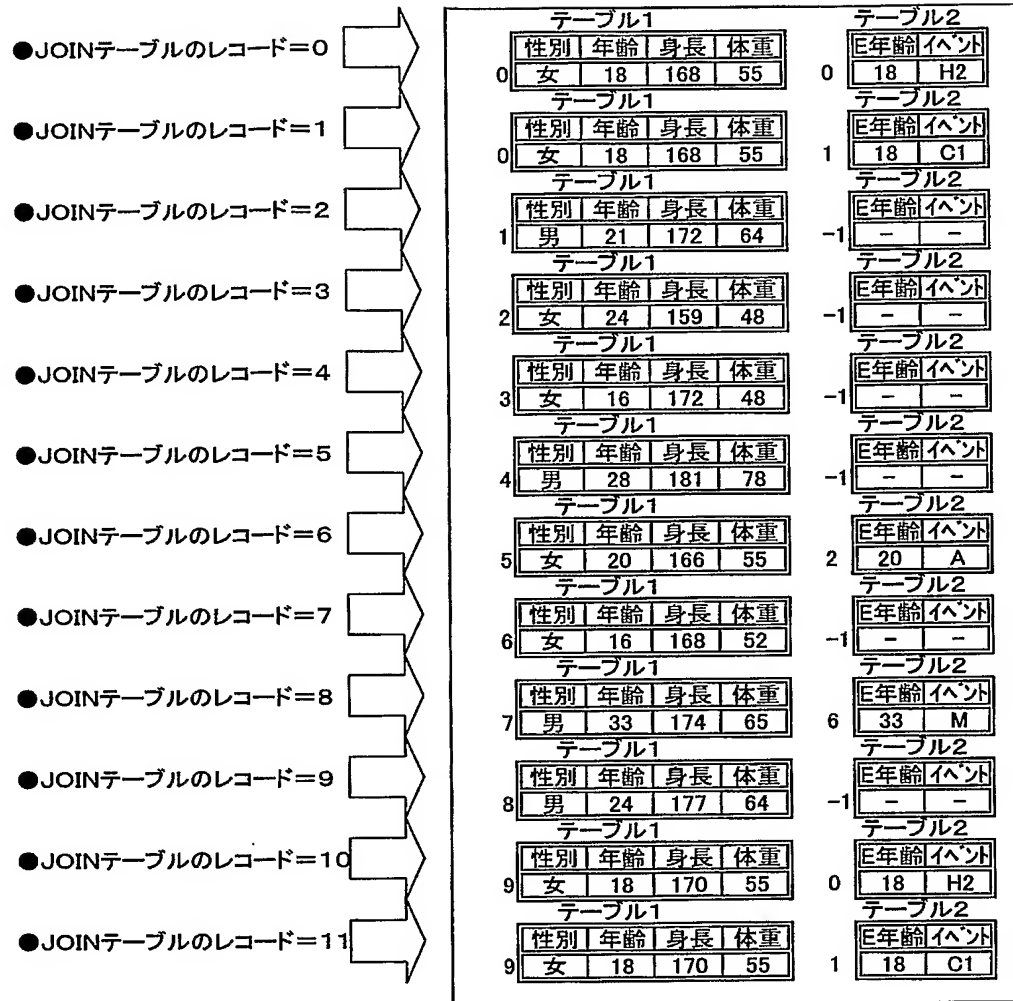


【図 7 6】



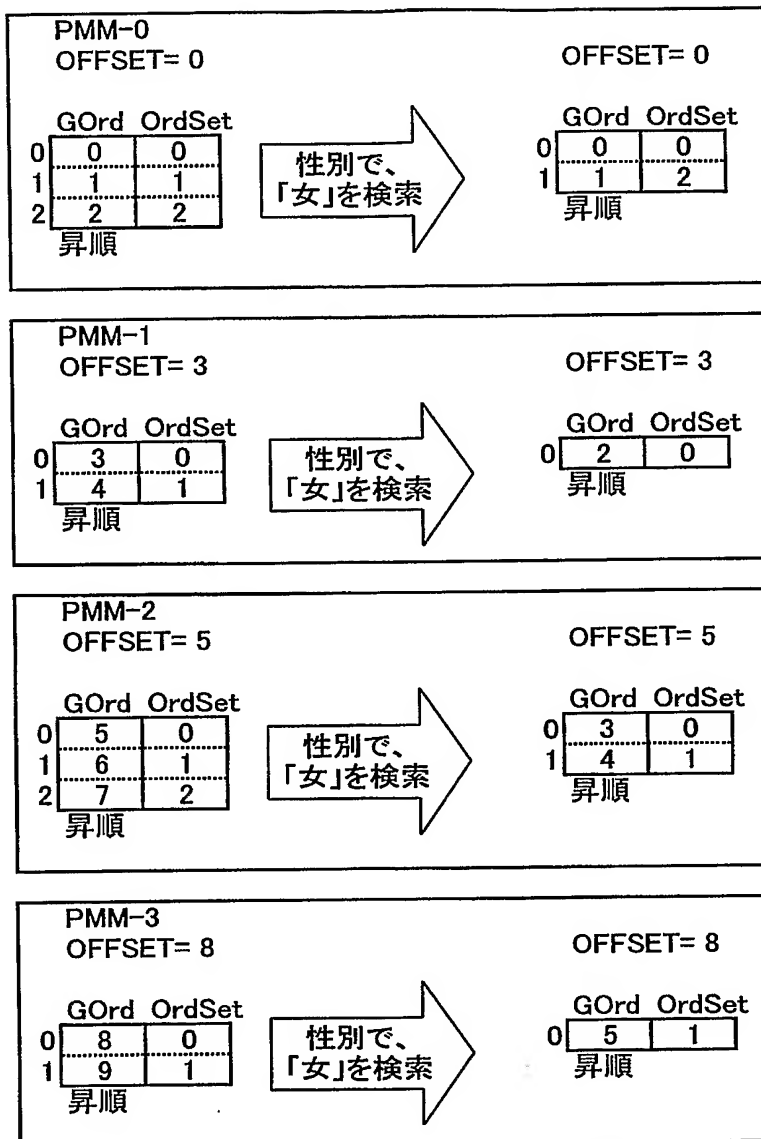
【図 78】

図78

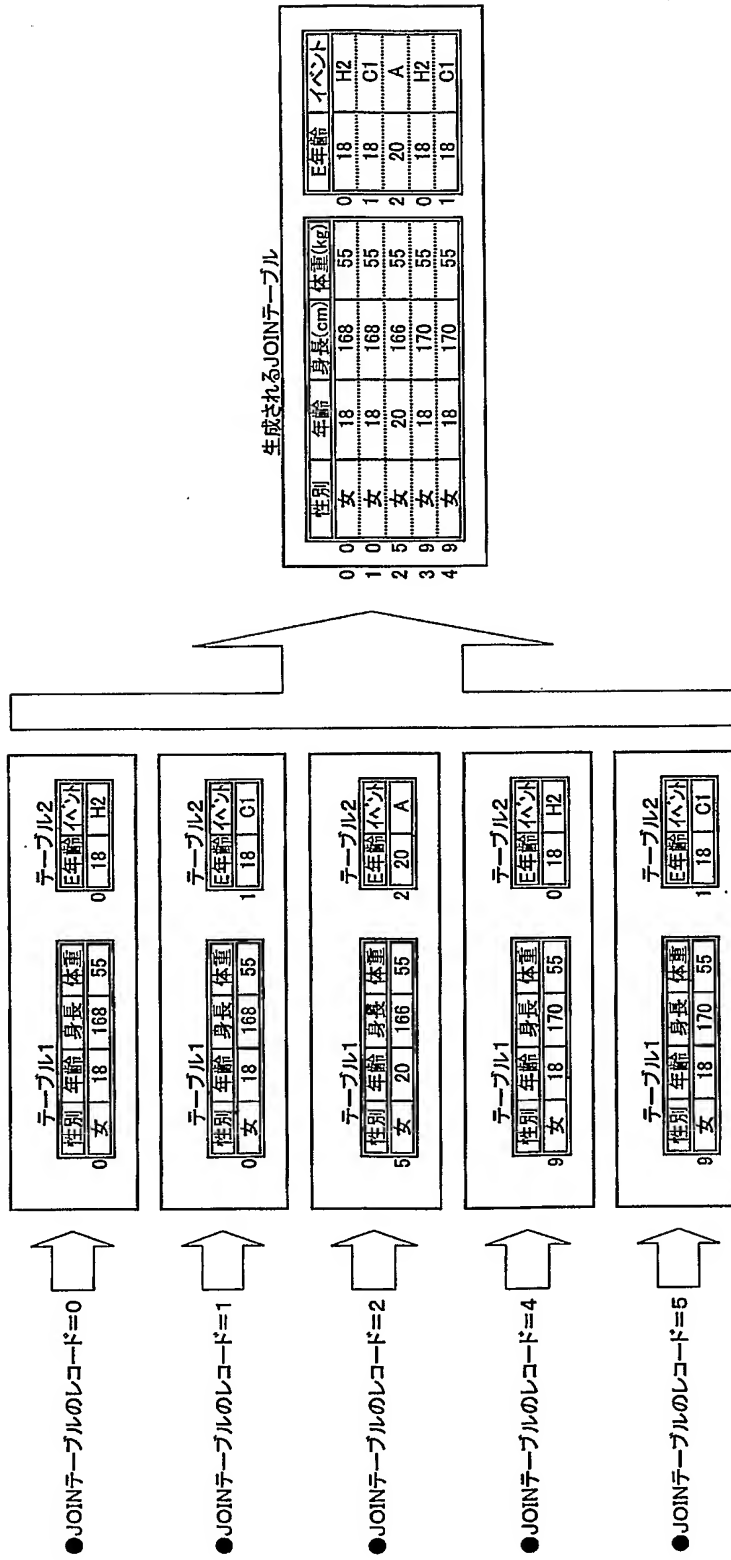


【図 79】

図79

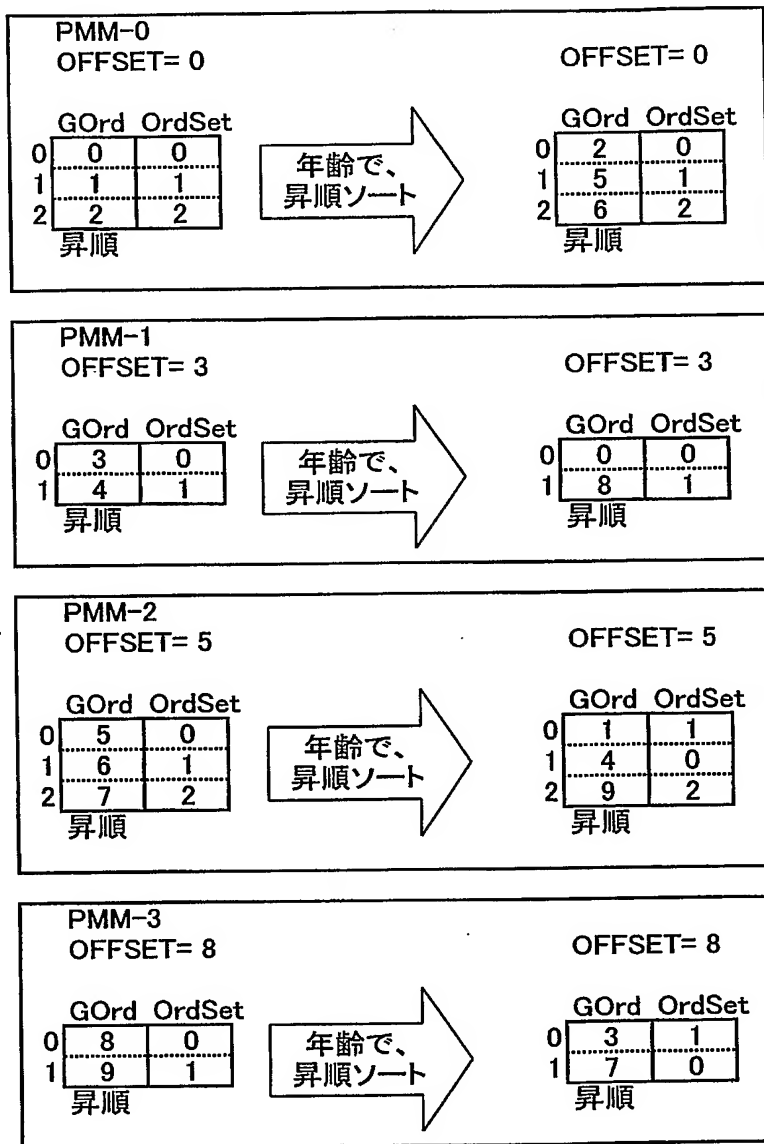


【図 80】



【図 81】

図81



【図 82】

図 82

PMM-0

性別	年齢	身長	体重
0 女	18	168	55
1 男	21	172	64
2 女	24	159	48

OFFSET=0

SetAggr	GO	Ord	Set
0	0	2	0
1	5	5	1
2	5	6	2

昇順 昇順

PMM-1

性別	年齢	身長	体重
0 女	16	172	48
1 男	28	181	78

OFFSET=3

SetAggr	GO	Ord	Set
0	0	0	0
1	5	8	1

昇順 昇順

PMM-2

性別	年齢	身長	体重
0 女	20	166	55
1 女	16	168	52
2 男	33	174	65

OFFSET=5

SetAggr	GO	Ord	Set
0	0	1	1
1	4	4	0
2	5	9	2

昇順 昇順

PMM-3

性別	年齢	身長	体重
0 男	24	177	64
1 女	18	170	55

OFFSET=8

SetAggr	GO	Ord	Set
0	2	3	1
1	5	7	0

昇順 昇順

年 齢 (PMM-0)

VNo	0	1	2
0	0	1	2
1	1	3	0
2	2	4	5

昇順 昇順 昇順

VL	0	1	2
0	18	1	2
1	21	3	0
2	24	4	5

昇順 昇順 昇順

GVNo	0	1	2
0	1	1	2
1	3	3	0
2	4	5	0

昇順 昇順 昇順

Count	0	1	2
0	2	0	3
1	0	0	5
2	0	0	0

昇順 昇順 昇順

年 齢 (PMM-1)

VNo	0	1
0	0	1

昇順

VL	0	1
0	16	28
1	0	0

昇順 昇順

GVNo	0	1
0	0	5
1	0	7

昇順 昇順

Count	0	1
0	0	6
1	0	0
2	0	0

昇順 昇順

年 齢 (PMM-2)

VNo	0	1	2
0	1	0	2

昇順

VL	0	1	2
0	16	20	33
1	0	2	6
2	2	8	1

昇順 昇順 昇順

GVNo	0	1	2
0	0	0	0
1	0	2	1
2	6	8	1

昇順 昇順 昇順

Count	0	1	2
0	0	2	6
1	0	1	2
2	0	0	0

昇順 昇順 昇順

年 齢 (PMM-3)

VNo	0	1
0	1	0

昇順

VL	0	1
0	18	24
1	1	1

昇順 昇順

GVNo	0	1
0	1	4
1	1	5

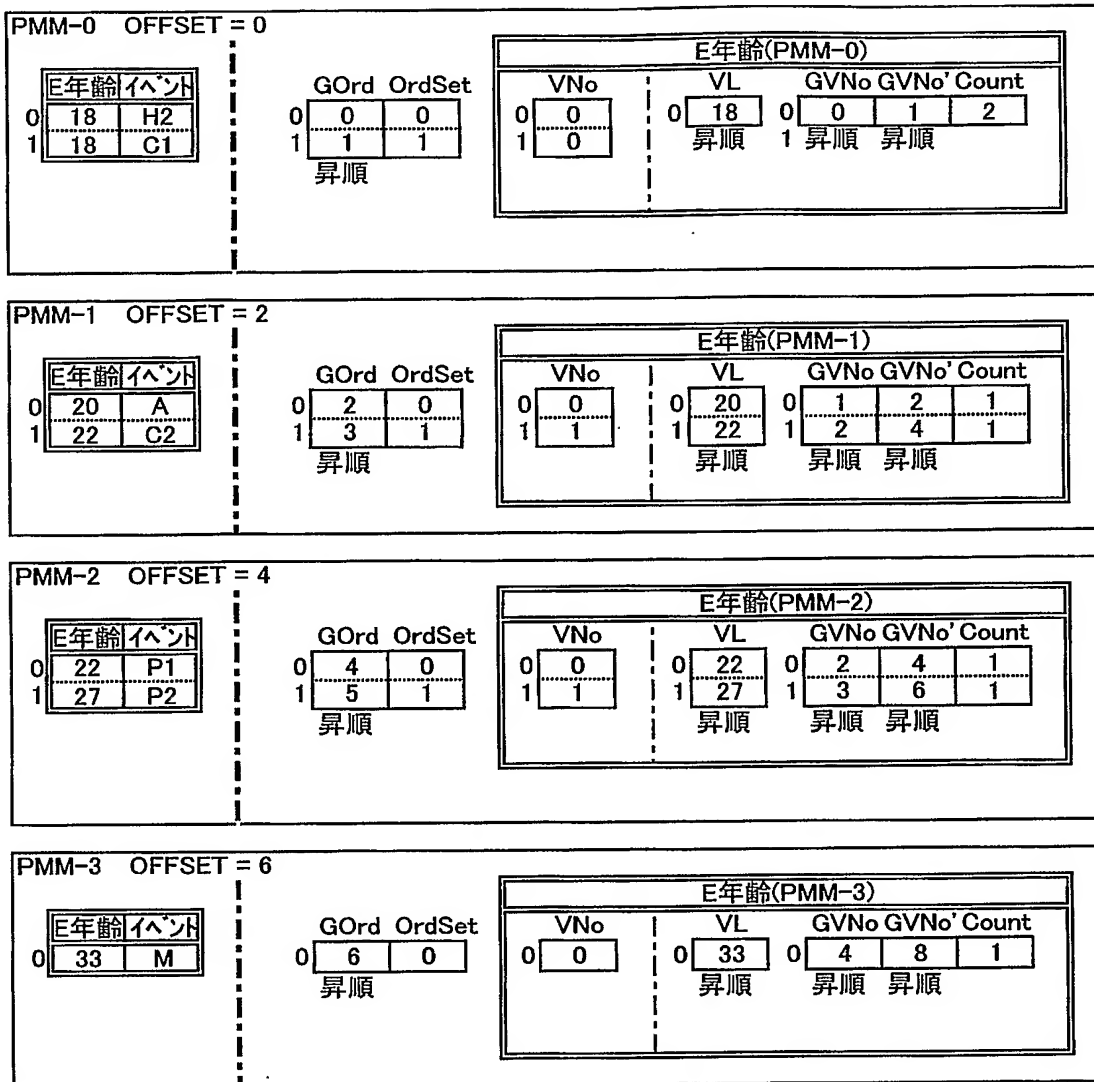
昇順 昇順

Count	0	1
0	2	0
1	2	0
2	0	0

昇順 昇順

【図 83】

図83



【図 84】

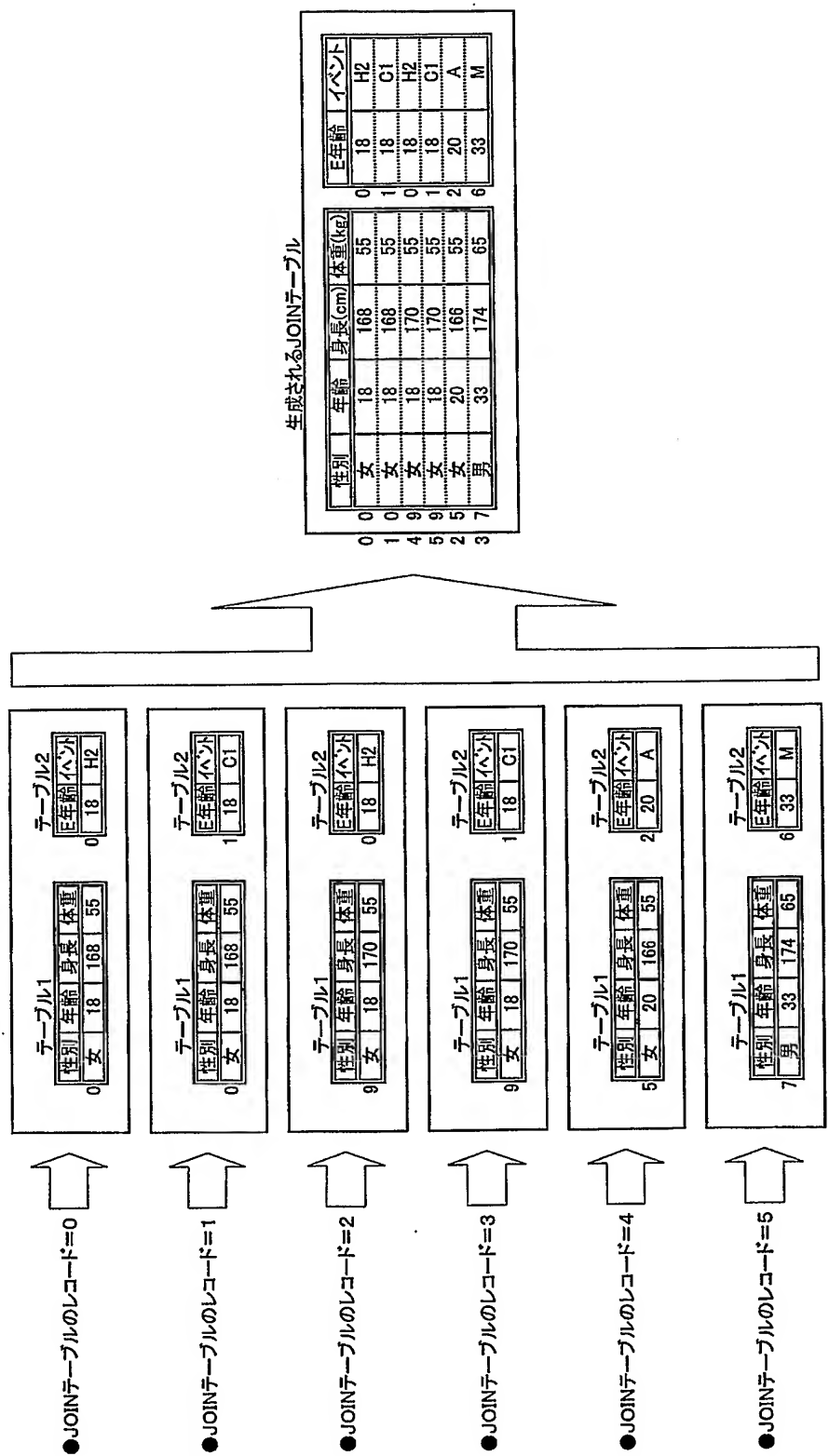
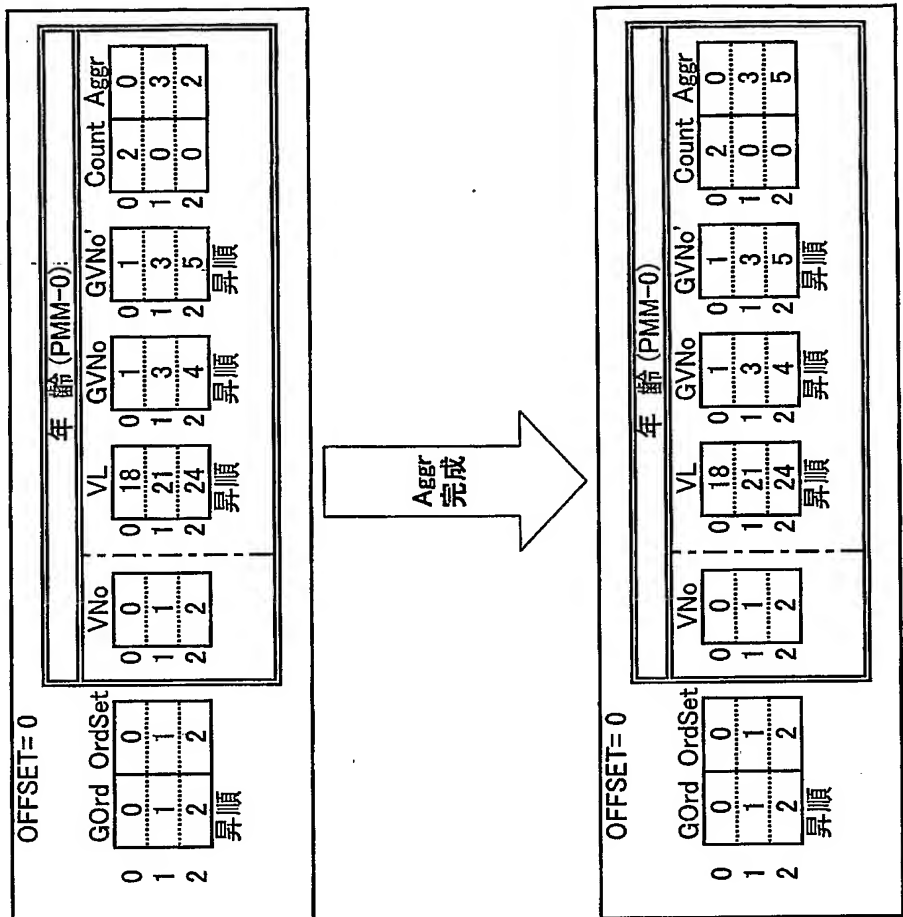


図 84

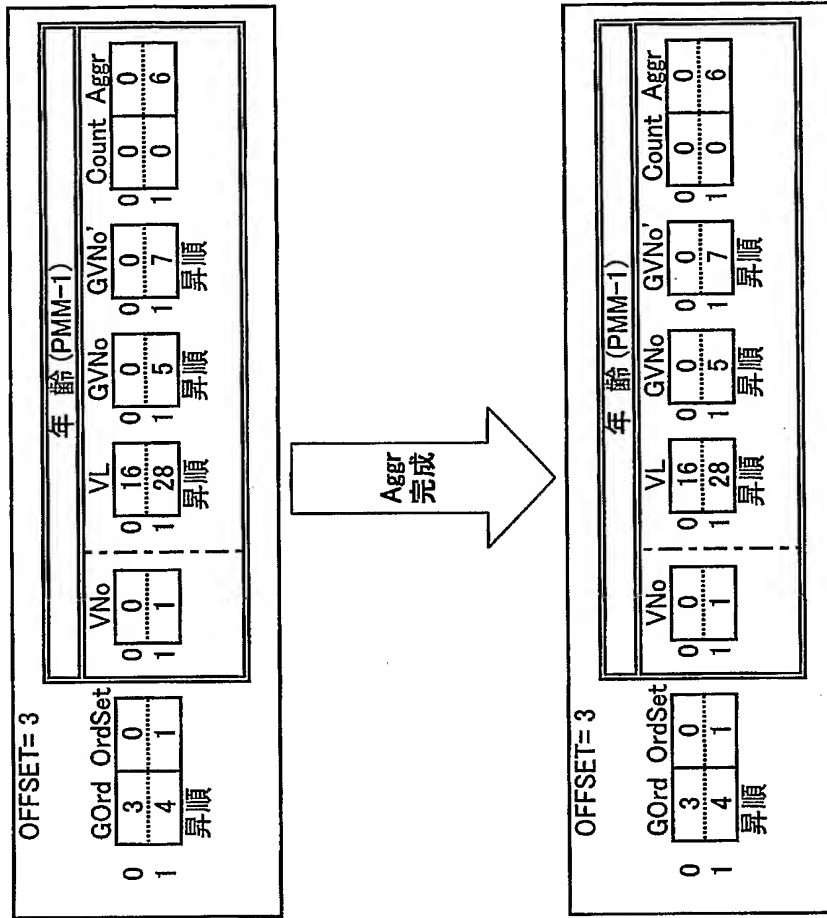
【図 86】

図86

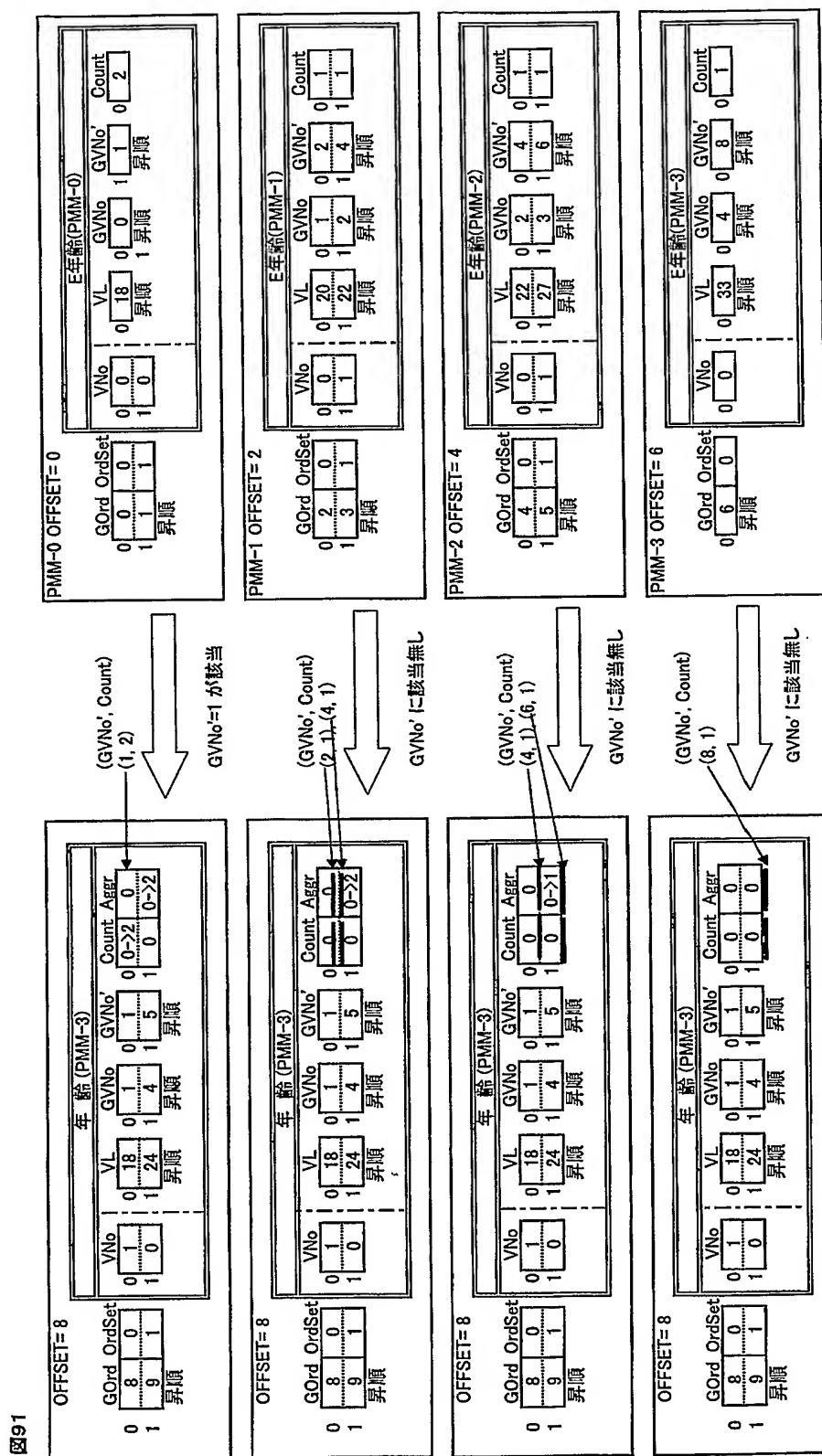


【図 88】

図88

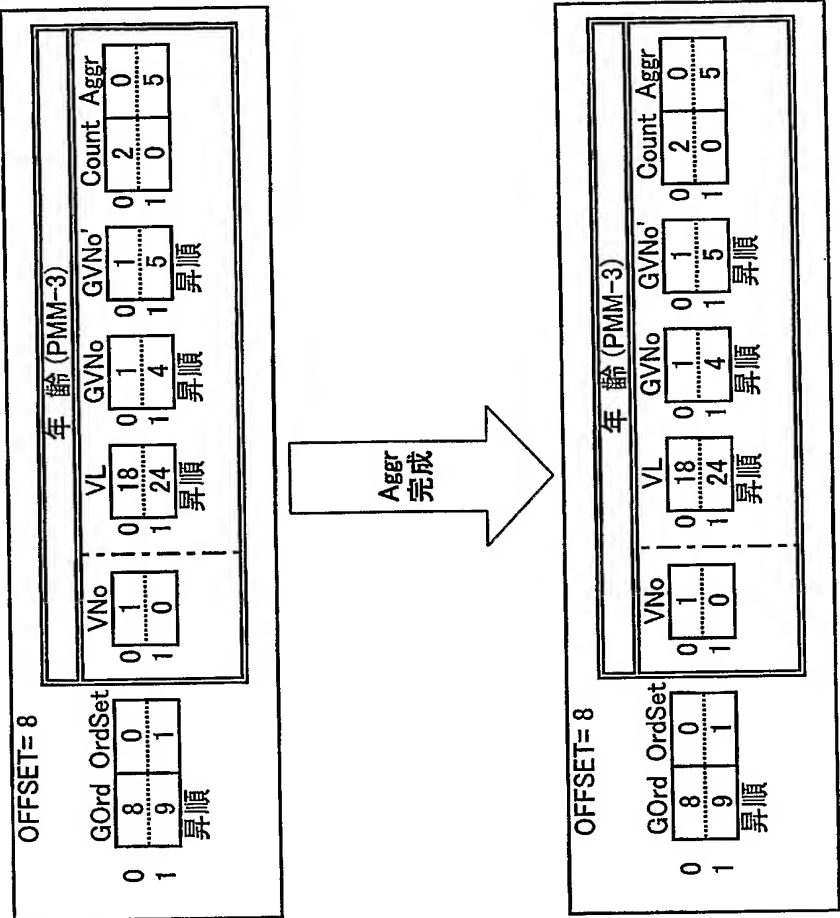


【図 9 1】



【図92】

図92



【書類名】 要約書

【要約】

【課題】 処理と通信を統合して高速な並列処理を実現し、高速に表形式データをジョインさせる。

【解決手段】 情報処理システムは、複数のPMM12と、PMM間を接続し、あるPMMの値を他のPMMに伝達するデータ伝送路14, 16とを備える。各PMMのメモリには、昇順または降順に重複なく順序付けられた、第1の項目の値のリストおよび/または共有化すべき第2の項目の値のリストが保持される。各PMMのメモリモジュールは、他のPMMに、値のリストに含まれる値を送信するとともに、他のPMMから値のリストに含まれる値を受信し、他のPMMの、第1の項目の値のリストおよび第2の項目の値のリストを参照して、他の全てのPMMの第1の項目および第2の項目の値のリストに含まれる値を考慮した、共通化された値のリストを生成する。

【選択図】 図1

認定・付加情報

特許出願の番号	特願 2 0 0 4 - 0 2 0 8 2 8
受付番号	5 0 4 0 0 1 4 5 8 3 5
書類名	特許願
担当官	伊藤 雅美 2 1 3 2
作成日	平成 1 6 年 1 月 3 0 日

<認定情報・付加情報>

【提出日】 平成16年 1月29日

特願 2 0 0 4 - 0 2 0 8 2 8

出 願 人 履 歴 情 報

識別番号 [5 9 8 1 0 8 5 1 5]

1. 変更年月日	1 9 9 8 年 8 月 1 1 日
[変更理由]	新規登録
住 所	神奈川県横浜市神奈川区松見町 4 丁目 1 1 0 1 番地 7 コート ハウス菊名 8 0 4 号
氏 名	古 庄 晋 二

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/000886

International filing date: 25 January 2005 (25.01.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-020828
Filing date: 29 January 2004 (29.01.2004)

Date of receipt at the International Bureau: 17 March 2005 (17.03.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse